FUJITSU SEMICONDUCTOR
CONTROLLER MANUAL

CM41-00312-3E

F²MC-16 FAMILY

SOFTUNE™ WORKBENCH

OPERATION MANUAL

FUJITSU

F^2MC-16 FAMILY

SOFTUNE™ WORKBENCH

OPERATION MANUAL

FUJITSU LIMITED

PREFACE

■ What is the SOFTUNE WORKBENCH?

SOFTUNE WORKBENCH is support software for developing programs for the F^2MC-16 families of microprocessors / microcontrollers.

It is a combination of a development manager, simulator debugger, emulator debugger, monitor debugger, and an integrated development environment for efficient development.

Note: F^2MC is the abbreviation of FUJITSU Flexible Microcontroller.

■ Purpose of this manual and target readers

This manual explains how to operate the SOFTUNE Workbench and design the product.

This manual is intended for engineers designing several kinds of products using SOFTUNE Workbench.

■ Trademarks

SOFTUNE is a trademark of FUJITSU LIMIITED.

Microsoft and Windows are registered trademarks in the United States and other countries of United States Microsoft Corporation.

Other company names and brand names are the trademarks or registered trademarks of their respective owners.

■ Organization of Manual

This manual consists of five chapters.

CHAPTER 1 OUTLINE OF SOFTUNE WORKBENCH

This chapter gives an outline of SOFTUNE WORKBENCH.

CHAPTER 2 OPERATION

This chapter describes the basic operation method and development procedure of SOFTUNE WORKBENCH.

CHAPTER 3 Windows

This chapter explains SOFTUNE WORKBENCH windows in detail.

CHAPTER 4 MENUS

This chapter explains in detail the SOFTUNE WORKBENCH menu configuration and the dialog boxes to be started from each menu.

CHAPTER 5 Addin Module

This chapter explains SOFTUNE WORKBENCH Addin module.

Appendix

The Appendixes describes the register name, downloading monitor program, setting LAN interface, setting USB interface, creating ROM on monitor debugger target, display on emulator, and external I/F DLL for simulator.

- The contents of this document are subject to change without notice.
  Customers are advised to consult with FUJITSU sales representatives before ordering.
- The information, such as descriptions of function and application circuit examples, in this document are presented solely for the purpose of reference to show examples of operations and uses of Fujitsu semiconductor device; Fujitsu does not warrant proper operation of the device with respect to use based on such information. When you develop equipment incorporating the device based on such information, you must assume any responsibility arising out of such use of the information. Fujitsu assumes no liability for any damages whatsoever arising out of the use of the information.
- Any information in this document, including descriptions of function and schematic diagrams, shall not be construed as license of the use or exercise of any intellectual property right, such as patent right or copyright, or any other right of Fujitsu or any third party or does Fujitsu warrant non-infringement of any third-party' s intellectual property right or other right by using such information. Fujitsu assumes no liability for any infringement of the intellectual property rights or other rights of third parties which would result from the use of information contained herein.
- The products described in this document are designed, developed and manufactured as contemplated for general use, including without limitation, ordinary industrial use, general office use, personal use, and household use, but are not designed, developed and manufactured as contemplated (1) for use accompanying fatal risks or dangers that, unless extremely high safety is secured, could have a serious effect to the public, and could lead directly to death, personal injury, severe physical damage or other loss (i.e., nuclear reaction control in nuclear facility, aircraft flight control, air traffic control, mass transport control, medical life support system, missile launch control in weapon system), or (2) for use requiring extremely high reliability (i.e., submersible repeater and artificial satellite).
  Please note that Fujitsu will not be liable against you and/or any third party for any claims or damages arising in connection with above-mentioned uses of the products.
- Any semiconductor devices have an inherent chance of failure. You must protect against injury, damage or loss from such failures by incorporating safety design measures into your facility and equipment such as redundancy, fire protection, and prevention of over-current levels and other abnormal operating conditions.
- If any products described in this document represent goods or technologies subject to certain restrictions on export under the Foreign Exchange and Foreign Trade Law of Japan, the prior authorization by Japanese government will be required for export of those products from Japan.

Copyright© 2004-2007 FUJITSU LIMITED All rights reserved

READING THIS MANUAL

■ Configuration of Page

In each section of this manual, the summary about the section is described certainly, so you can grasp an outline of this manual if only you read these summaries.

And the title of upper section is described in lower section, so you can grasp the position where you are reading now.

■ Product name abbreviation

In this manual and this product, product names are abbreviated as follows:

The Microsoft® Windows® 2000 Professional operating system is abbreviated to Windows 2000.

The Microsoft® Windows® XP Professional operating system is abbreviated to Windows XP.

CONTENTS

CHAPTER 1 OUTLINE OF SOFTUNE WORKBENCH 1

1.1 Outline 2
1.2 What is SOFTUNE WORKBENCH? 3
1.3 Procedure for Developing Programs with SOFTUNE WORKBENCH 4

CHAPTER 2 OPERATION 5

2.1 Parameters to be Entered from Dialog Boxes 6
2.1.1 Data and Address Formulas (Numerical constant) 7
2.1.2 Data and Address Formulas (Symbols, Line Numbers, Character Constants) 9
2.1.3 Data and Address Formulas (Register name, Flag name) 11
2.1.4 Operators Usable in Data and Address Formulas 12
2.1.5 Address Formula Specification 14
2.1.6 File Name Specification 15
2.2 Starting and Terminating SOFTUNE WORKBENCH 16
2.3 Creating Workspace 17
2.4 Storing Project 19
2.5 Creating and Registering Source File in Project 21
2.6 Definition of Subproject 22
2.7 Creation of Project Configuration 23
2.8 Setting Tools 25
2.9 Setting Linker Options 26
2.10 Make/Build 28
2.10.1 Making or Building of Project 29
2.11 Debugging 30
2.12 Executing Debugging Only 31
2.13 Reading SOFTUNE Project Files of Old Versions 33
2.14 Moving Project Files 35
2.15 Useful Functions 36

CHAPTER 3 Windows 37

3.1 Window Configuration 39
3.2 Tool Bar 40
3.3 Status Bar 43
3.4 Project Window 44
3.4.1 SRC tab 45
3.4.2 ABS tab 56
3.5 Output Window 60
3.6 Edit Window (Standard Editor) 62
3.6.1 Setting Standard Editor 66
3.7 Source Window 70
3.8 Symbol Window 74
3.9 Disassemble Window 76
3.10 Register Window 79

3.10.1 Setting Register Display 80
3.11 Memory Window 82
3.12 Local Variable Window 84
3.13 Watch Window 85
3.14 Trace Window 87
3.15 Command Window 89
3.15.1 Setting Character String Replacement 91
3.15.2 Logging 94
3.16 Object Window 95
3.17 Coverage Window 97
3.18 Performance Window 99
3.19 Sequence Window 102
3.20 Real-time Memory Window 103
3.21 RAM Checker Window 104
3.21.1 Setup of the RAM Checker 106
3.21.2 Start the RAM Checker viewer 109

CHAPTER 4 MENUS 111
4.1 Menu Configuration (Tree) 112
4.2 File Menu 115
4.2.1 New 116
4.2.2 Open 119
4.2.3 Close 121
4.2.4 Open Workspace 122
4.2.5 Close Workspace 123
4.2.6 Save 124
4.2.7 Save As 125
4.2.8 Save All 127
4.2.9 Print 128
4.2.10 Recent Text File/Recent Workspace File 129
4.2.11 Exit 130
4.3 Edit Menu 131
4.3.1 Undo, Redo 132
4.3.2 Cut, Copy, Paste, Delete 133
4.3.3 Select All 135
4.3.4 Find/Replace 136
4.3.5 Find in Files 138
4.3.6 Jump 139
4.3.7 Bookmark 140
4.3.7.1 Bookmark - Source window 141
4.3.7.2 Bookmark - Memory window 144
4.3.7.3 Bookmark - Edit window 146
4.3.8 Previous error, Next error, Top of Error, Bottom of Error 148
4.3.9 Property 149
4.3.9.1 Property-Project Window 150
4.3.9.2 Property-Edit Window 154
4.3.9.3 Property-Source Window 155

4.3.9.4 Property-Symbol Window 156
4.3.9.5 Property-Local Window 157
4.3.9.6 Property-Watch Window 158
4.4 View Menu 159
4.4.1 Project/Output 160
4.4.2 Symbol 161
4.4.3 Assembly 162
4.4.4 Register 164
4.4.5 Memory 166
4.4.6 Local 175
4.4.7 Watch 176
4.4.8 Trace 178
4.4.9 Command 202
4.4.10 Tool Bar, Status Bar, Tab 203
4.4.11 Object 204
4.4.12 Coverage 210
4.4.13 Performance 212
4.4.14 Fonts 215
4.4.15 Real-time Memory 216
4.4.16 RAM Checker 218
4.5 Project 219
4.5.1 Active Project 220
4.5.2 Add Project 221
4.5.2.1 Add Project - Create 222
4.5.2.2 Add Project - Existing Project 224
4.5.3 Add Member 225
4.5.3.1 Add Member - File 226
4.5.3.2 Add Member - Directory 227
4.5.4 Setup Workspace 228
4.5.5 Setup Project 229
4.5.5.1 General 232
4.5.5.2 MCU 234
4.5.5.3 Setting C Compiler Options 244
4.5.5.4 Setting Assembler Options 251
4.5.5.5 Setting Linker Options 256
4.5.5.6 Section Disposition/Connection Specifying 262
4.5.5.7 Setting Librarian Options 267
4.5.5.8 Setting Converter Options 269
4.5.5.9 Setting Debug Options 272
4.5.6 Setting Customize Build 275
4.5.7 Project Dependencies 279
4.5.8 Project Configuration 280
4.5.8.1 Project Configuration - Add and Delete 281
4.5.8.2 Project Configuration - Configuration at Build 283
4.5.9 Include Dependencies 284
4.5.10 Compile, Make, Build, and Stop 285
4.6 Debug 286

4.6.1 Run 287
4.6.2 Abort 288
4.6.3 Reset MCU 289
4.6.4 Break Point 290
4.6.4.1 Breakpoint Details Setting 298
4.6.5 Event 302
4.6.6 Sequence 307
4.6.7 Stack 312
4.6.8 Time Measurement 313
4.6.9 Call 314
4.6.10 Clear Call 316
4.6.11 Vector 317
4.6.12 Load Target File 318
4.6.13 Start Debug/Terminate Debug 319
4.7 Setup 320
4.7.1 Development 321
4.7.2 Debug Environment 325
4.7.2.1 I/O Port 326
4.7.2.2 Interrupt 330
4.7.2.3 Debug Environment 331
4.7.2.4 Selection Debug Function 350
4.7.2.5 Setup Wizard 351
4.7.3 Memory Map 353
4.7.4 Tool 359
4.7.5 Keyboard 362
4.7.6 Editor 364
4.7.7 Error 366
4.7.8 Tool Startup 368
4.8 Window 369
4.8.1 Cascade, Vertical, Horizon 370
4.8.2 Split 371
4.8.3 Arrange Icons 372
4.8.4 Refresh Window 373
4.8.5 Refresh All Windows 374
4.8.6 Close All Windows 375
4.9 Help 376
4.9.1 Help Topics 377
4.9.2 Support Information 378
4.9.3 About Fs907s... 379

CHAPTER 5 Addin Module 381
5.1 Customize Bar 382
5.1.1 What is Customize Bar? 383
5.1.2 Customize Bar Menu 385
5.1.3 Registering in the Customize Bar 386
5.1.3.1 Registering Batch File 390
5.1.3.2 Registering Workbench Menu 392

5.1.3.3 Registering External Tool 394
5.1.4 Warning and Error Messages 396
5.1.5 Note 398

APPENDIX 399
APPENDIX A List of Register Names 400
APPENDIX B Downloading Monitor Program 401
APPENDIX C Setting LAN Interface 403
APPENDIX D Setting USB Interface 405
APPENDIX E Creating ROM on Monitor Debugger Target 406
APPENDIX F Building REALOS Debug Modules into Program 410
APPENDIX G Display on Emulator 411
APPENDIX H Guide to Changing Function-call Interface 413
APPENDIX I External I/F DLL for Simulator 421

CHAPTER 1

OUTLINE OF SOFTUNE WORKBENCH

SOFUTUNE WORKBENCH integrates language tools and debuggers into one to provide the integrated development environment that totally supports processing from programming and debugging to creation of data to be written to ROM. Language tools include a C compiler, assembler, and linkage tool, etc. Debuggers are a simulator debugger, emulator debugger, and monitor debugger.

1.1 Outline

1.2 What is SOFTUNE WORKBENCH?

1.3 Procedure for Developing Programs with SOFTUNE WORKBENCH

1.1 Outline

This section gives an outline of the development tools integrated by SOFTUNE WORKBENCH.

■ Language tools

In the past, language tools (e.g., C compiler, assembler, and linkage kit) were started and used from command lines.

However, SOFTUNE WORKBENCH can use these tools as they area. An option setting dialog box for each tool opens, thereby enabling the easy use of the tools

■ Debuggers

SOFTUNE WORKBENCH has integrated the simulator debugger, emulator debugger, and monitor debugger into one. The optimum debugger can be selected and used as required.

■ Others

Installing an REALOS configurator (option) enables cooperative operation without complicated setting.

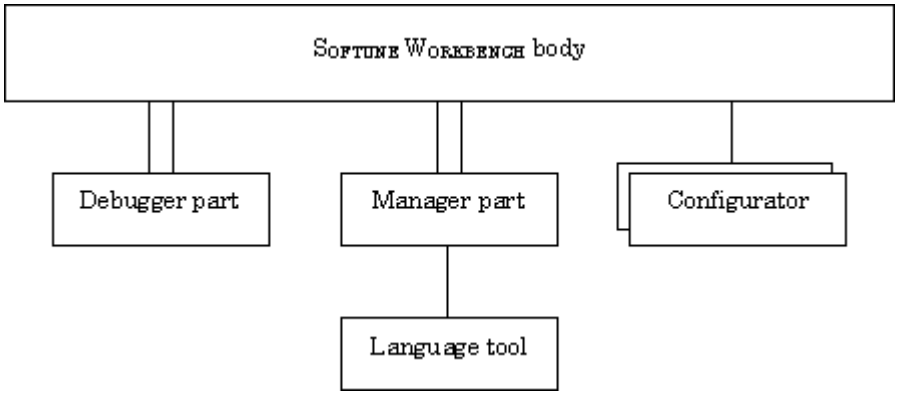
1.2 What is SOFTUNE WORKBENCH?

This section explains the basic configuration of SOFTUNE WORKBENCH.

■ SOFTUNE WORKBENCH configuration

Figure 1.2-1 shows the basic configuration of SOFTUNE WORKBENCH.

Figure 1.2-1 Basic Configuration of SOFTUNE WORKBENCH



As shown in Figure 1.2-1 SOFTUNE WORKBENCH consists of three parts: body, debugger, and manager.

The debugger part contains the simulator debugger, emulator debugger, and monitor debugger. These debuggers can be switched and used as required.

The manager part enables users to code and make programs without full knowledge of language tool (e.g., C compiler and assembler) start and option specification.

The configurator is not built into SOFTUNE WORKBENCH because it is an option. Installing this option, however, enables cooperative operation on SOFTUNE WORKBENCH.

SOFTUNE WORKBENCH manages all processing from programming to debugging in units of projects. Projects contain all program files, options of tools (e.g., C compiler), and debugger environment setup, etc.

1.3 Procedure for Developing Programs with SOFTUNE WORKBENCH

The procedure for developing programs with SOFTUNE WORKBENCH consists of the followings:

1. **Setting SOFTUNE WORKBENCH operating conditions**
2. **Creating a project**
3. **Creating a program source and executing make/build**
4. **Using the debugger**

■ Setting SOFTUNE WORKBENCH operating conditions

When developing a program with SOFTUNE WORKBENCH, first open the development environment setup dialog box from the [Setup] - [Development] Menu and set environment variables and projects. For details about how to set environment variables and projects, see Section "4.7.1 Development".

The environment variables set from this dialog box are referenced by language tools such as the C compiler.

■ Creating a project

Set information for the program to be developed in a project.

When developing a new project, open the new creation dialog box from the [File] - [New] Menu and select [Workspace/Project File] from the dialog box. The new project creation dialog box opens.

When the project has already existed, the existing project file can be opened from the [File] - [Open Workspace] Menu. When using the SOFTUNE V01 or V02 project file, see Section "2.13 Reading SOFTUNE Project Files of Old Versions".

■ Creating a program source and executing make/build

Open the new creation dialog box from the [File] - [New] Menu and select [Text File]. When the editor is started, write the source program and save it to the file with the [File] - [Save As] Menu.

When a necessary source file is created, register it in the project with the [Project] - [Add Member] Menu.

When registering the source file in the project is completed, execute "make" with the [Project] - [Make] Menu or execute "build" with the [Project] - [Build] Menu.

If a syntax error occurs during compilation or assembling, double-click the error display location in the Output Window with the left button of the mouse. The program jumps to the line where the error occurred. Correct the source file, and then reexecute the [Project] - [Make] Menu.

■ Using the debugger

When a load module file is created, debugging can be begin.

CHAPTER 2
OPERATION

This chapter explains the basic operation of SOFTUNE WORKBENCH for each of the following items:

2.1 Parameters to be Entered from Dialog Boxes

2.1.1 Data and Address Formulas (Numerical constant)

2.1.2 Data and Address Formulas (Symbols, Line Numbers, Character Constants)

2.1.3 Data and Address Formulas (Register name, Flag name)

2.1.4 Operators Usable in Data and Address Formulas

2.1.5 Address Formula Specification

2.1.6 File Name Specification

2.2 Starting and Terminating SOFTUNE WORKBENCH

2.3 Creating Workspace

2.4 Storing Project

2.5 Creating and Registering Source File in Project

2.6 Definition of Subproject

2.7 Creation of Project Configuration

2.8 Setting Tools

2.9 Setting Linker Options

2.10 Make/Build

2.10.1 Making or Building of Project

2.11 Debugging

2.12 Executing Debugging Only

2.13 Reading SOFTUNE Project Files of Old Versions

2.14 Moving Project Files

2.15 Useful Functions

2.1 Parameters to be Entered from Dialog Boxes

When key entry is requested from a dialog box, the following four elements can be written as parameters:
- Data formula
- Address formula
- Identifier
- File name specification

■ Data formula

A data formula consists of a term and an operator. Data formulas comply with C language formulas. Almost C language formulas are recognized. Some points (e.g, line number and register specification) are extended. Operations involving floating-point numbers and character strings are not supported. The overflows that occur during the operation are ignored. Zero division results in an error.

■ Address formula

An address formula is an extension of the data formula; it represents a memory location. Like the data formula, the address formula consists of a term and an operator. The terms and operators usable in address formulas are the same as those in data formulas

■ Identifier

Alphabetic characters, numbers, and "_" can be used as identifiers. Each identifier must begin with a character other than numbers. Uppercase characters are distinguished from lowercase characters or vice versa.

■ File name specification

File name specification complies with Windows rules.

2.1.1 Data and Address Formulas (Numerical constant)

The SOFTUNE WORKBENCH provides numeric constants as the terms of data and address formulas. An integer or floating-point number can be written as a numerical constant.

■ Integer

When writing an integer, specify a specifier (B', Q', D', H', 0x) representing the base number of the numerical value. If no specifier is specified, the base number specified by the default is used. The default base numbers are determined for each of the locations from which values must be entered. For details, see each dialog box.

The numerical value representation range is from 0 to H'FFFFFFFF.

However, this range is further restricted, depending on the values to be entered.

The minus values are represented such as D'1.

(Example)

Binary constant	B'1010
Octal constant	Q'1267
Decimal constant	D'1800
Hexadecimal constant	H'12AF or 0x12AF

Note:

No blank is allowed between a specifier and a numerical value.

■ Floating-point number

The following two floating-point numbers are supported.

- Single-precision floating-point number (S)....float
- Double-precision floating-point number (D)....double, long double

The internal format and size comply with the floating-point type handled by the C compiler.

```
[F'][ ]{.d|d[.[d]]}[{S|D}[[+| -]d]]
```

"d" specifies an unsigned decimal number

Nearest value rounding applies to input values. If the represented value is not a normalized number, a warning message is displayed and the following value is input:

- When an underflow occurs.... The values that can be represented as unnormalized numbers are changed to unnormalized numbers. The values less than unnormalized numbers are changed to ± 0.
- When an overflow occurs.... Values are changed to infinity.

A floating-point number can also be specified in a hexadecimal number as follows:

```
H' hexadecimal - number [.{S|D}]
```

Note:

If S and D are omitted, D is assumed.

2.1.2 Data and Address Formulas (Symbols, Line Numbers, Character Constants)

The SOFTUNE WORKBENCH provides symbols, line numbers, and character constants as the terms of data and address formulas.

■ Symbol

The symbols used in the source program can be referenced as addresses; they have the type information generated by the C compiler and other accessory information. The accessory information generated by the assembler is label information. Each symbol consists of a module name, a function name, and a symbol name. Specify these names as follows:

```
[[module-name][ \function-name]\] symbol-name
```

When the source program is written in the assembly language, module-name is the name written in the operand of the [PROGRAM] statement. When it is written in the C language, module-name is the name of the source file to be compiled. Function-name is a function name written in the C language; it is valid only when the source program is written in the C language.

To distinguish a global symbol from others, write it as \symbol-name.

Table 2.1-1 shows symbol description.

No Japanese character strings can not be used for symbols.

Table 2.1-1 Symbol Cannot Description

Description	Contents
Symbol	Local symbol in function → static symbol in module → global symbol Global symbol
\Symbol	Local symbol in the specified function in the current module
\function\symbol	Static symbol in the specified module or global symbol defined in the specified module
Module\symbol	Local symbol in the specified function in the specified module
Module\function\symbol	Symbol in class (Valid for static) Symbol in class function

■ Line number

The line numbers to be generated by the C compiler or assembler can be used to reference addresses. For the C language, a line number indicates the starting address when one line is compiled.

```
[source file-name]$line-number
```

When referencing an address with a line number, prefix $ to the line number. A line number can be specified only in a decimal number. Specify the line number in the following format. If the extension of the source file name is .c, line number specification can be omitted. If the source file name is not alphanumeric characters, enclose the line number in double quotes.

■ Character constant

A character constant is the character value enclosed in a single quote; it cannot include a single quote and \. Instead of these characters (single quote and back slash), escape characters can be used as character constants. Characters that can constitute character strings can be used as escape characters.

2.1.3 Data and Address Formulas (Register name, Flag name)

The SOFTUNE WORKBENCH provides register and flag names as the terms of data and address formulas.

■ Register name, flag name

Register and flag names can be specified in data formulas; they represent the register values at that point of time.

Specify the register name and flag name, following %.

Usable register names differ for each MCU; see Appendix A Register Names .

2.1.4 Operators Usable in Data and Address Formulas

Table 2.1-2 lists the operators that can be used in data and address formulas and their priorities.

■ Operators Usable in Data and Address Formulas

Table 2.1-2 List of Operators Usable in Data and Address Formulas

Priority	Symbol	Explanation	Type of operator
1	( ) [ ] . - >	Priority change Subscript representation Structure Structure pointer	Linear expression
2	- & ! ~ * sizeof (type)	Minus sign Address Logical NOT Bit NOT Memory indirect reference Size (byte) Type cast	Binary operator
3	* / %	Multiplication Division Remainder	Binary operator
4	+ -	Addition Subtraction	
5	<< >>	Left shift Right shift	
6	< <= > >=	Less than Less than or equal to Greater than Greater than or equal to	Binary operator (Relational operator)
7	== !=	Equal to Not equal to	
8	&	Bit AND	Binary operator
9	^	Bit EOR	
10	|	Bit OR	
11	&&	Logical AND	
12	||	Logical OR	

Remarks:

When the comparison result is true, the relational operator becomes H'1. When false, it becomes H'0. The SOFTUNE WORKBENCH does not support the conditional operator (?:), comma operator (,), increment operator (++) and decrement operator (--) of C language.

2.1.5 Address Formula Specification

Address formula specification is divided into the Full addressing and Address range specification.

■ Full addressing

The full addressing format is as follows:

32-bit-addressing

32-bit-addressing : ....Expression for addressing

■ Address range specification

Address range specification, representing a memory range, consists of two addressings; it has any of the following two formats:

Addressing..addressing (from starting address to ending address) Addressing..+ offset (from starting address to starting address + offset)

offset : Value relative to addressing (starting address)

■ Bit addressing

The notation below is used to represent a bit address. Symbols for bit address attributes can be also used. Bit addressing is valid when /BIT is specified in the command qualifier.

[addressing] : bit-offset

bit-offset: ....Value used to specify a bit position

When addressing is omitted, address 0 is assumed.

■ Bank addressing

The bank addressing format is as follows:

bank-specification : offset-specification

bank-specification: Expression used to specify a bank address

offset-specification: Expression used to specify a 16-bit address

2.1.6 File Name Specification

File name specification complies with Windows for host environment.

■ File name specification

[drive-name:] [directory-path-name] file-name [.extension]

When drive-name is omitted, the current drive is selected.

2.2 Starting and Terminating SOFTUNE WORKBENCH

This section explains how to start and terminate SOFTUNE WORKBENCH.

■ Starting and terminating SOFTUNE WORKBENCH

With SOFTUNE V3, to start SOFTUNE WORKBENCH, double-click the [FFMC-16 Family Softune Workbench] icon in the [Softune V3] group.

When this program is started for the first time with SOFTUNE Manager V01 or V02 installed, the dialog box is displayed which asks whether or not to take over information for [Setting editor], [Setting tools], and [Setting error jump] set in the previous version. To take over the information, click the [Yes] button. Not to take over it, click the [No] button.

■ Terminating SOFTUNE WORKBENCH

To terminate SOFTUNE WORKBENCH, select [Exit] from the [File] Menu or click the x button above and on the right of the window.

SOFTUNE WORKBENCH cannot be terminated when compile/assemble, make, build, or tool is being executed. Be sure to terminate SOFTUNE WORKBENCH after compile/assemble, make, build, or tool has been terminated or suspended.

2.3 Creating Workspace

Workspace is necessary to store projects in SOFTUNE WORKBENCH to register project, follow the steps below.

■ Creation of Workspace

SOFTUNE WORKBENCH uses the following methods to create workspace.

- Creating workspace in creating new project

  When a new project is created, workspace to store the project is also created.

  In this case, the name and position of workspace are the same as those of the project.

- Creating workspace in opening project

  When [File]-[Open Workspace] is selected to open a project, workspace to store the project is also created.

  In this case, the name and position of workspace are the same as those of the project.

  If any workspace file already exists, however, it is opened instead of the project file.

- Creating blank workspace

  Blank space that has no project is created. Projects must be stored separately.

  In this case, projects can be stored different in name and position from workspace.

■ Creating Workspace in Creating New Project

1. Select [File]-[New].
   When the [New] dialog is opened, select "Workspace/Project File" in [Type of File] and click the [OK] button.
2. Select the [Project] tab.
   When the [New] dialog is opened, select the [Project] tab.
3. Select [Create New Workspace].
   Click the [Create New Workspace] check button to create workspace. Create workspace in the same way that a new project is created.

■ Creating Workspace in Opening Project

1. Select [File]-[Open Workspace].
2. Select "Project File" in [Type of File].
   When the [Open Workspace] dialog is opened, select "Project File" in [Type of File].
3. Select the project file to be opened.
   Select the project file to be opened.
4. Click the [Open] button.

■ Creating Blank Workspace

1. Select [File]-[New].
   When the [New] dialog is opened, select "Workspace/Project File" in [Type of File] and click the [OK] button.
2. Select the [Workspace] tab.
   When the [New] dialog is opened, select the [Workspace] tab.

3. Select the type of workspace.
   Select [Blank Workspace] as the type of workspace.
4. Enter the workspace name.
   Enter the workspace name. This name is used as a workspace file name. At default it is also used as a workspace directory (the workspace directory can be changed).
5. Click the [OK] button.

2.4 Storing Project

A project is necessary to develop and debug software in SOFTUNE WORKBENCH. To store a project in workspace, follow the steps below.

■ Storing of Project

SOFTUNE WORKBENCH uses the following methods to store a project in workspace. To store a new project is as active project in workspace.

- Storing new project in currently opened workspace

  A new project is stored in currently opened workspace.

- Storing existing projects in currently opened workspace

  Existing projects are stored in currently opened workspace.

■ Storing New Project in Currently Opened Workspace

1. Open workspace to store a project.
   Please open workspace to be stored a project.
2. Select [Project]-[Add Project]-[New].
3. Select [Add to Current Workspace].
   Click the [Add to Current Workspace] check button.
4. Select [Project Type].
   Select the type of file last created in the project [Project Type]. Table 2.4-1 indicates the selectable project types and their explanation.
5. Selecting of function-call interface
   Select the function-call interface to be used in C Compiler.
   Table 2.4-2 shows selectable function-call interfaces and their description.
6. Select the chip type and target MCU.
   Selectable values are indicated in the top-down list. Select the chip and target MCU from the list.
7. Enter the project name.
   Enter the project name. This name can be use as a project file name. At default it is also used as a target file name and project directory (the target file name and project directory can be changed).
8. Set project dependence.
   When defining a project as a subproject in another project, place a checkmark in the [Project Dependence] checkbox and select the project name from the list in the [Project Dependence] checkbox.
9. Click the [OK] button.

"Debug", "Debug\ABS", "Debug\OBJ", "Debug\LST", or "Debug\OPT" directory is created as a subdirectory in the project directory.

Debug: A directory to store information for each project configuration. The default configuration name as a new project is created is "Debug".

ABS:	Directory in which the target file is stored
OBJ:	Directory in which the object file is stored
LST:	Directory in which the list file is stored
OPT:	Directory in which the option file is stored

When REALOS is selected as the project type, the Setup Wizard of the configurator opens. For details, refer to the manual accompanying SOFTUNE REALOS.

Table 2.4-1 List of Project Types

Project type	Explanation
Absolute format (ABS)	An ordinary program file is created.
Relative format (REL)	A relative format file is created.
Library file	A library file is created.
REALOS (ABS)	A program that uses a real-time operating system is created.
Arguments by the stack	Passes function arguments by the stack. This is a default setting.
Arguments by registers	Passes selected function arguments by registers.These processing's reduce the code size and stack usage to improve the program execution speed.

Table 2.4-2 List of Function-call interface

Function-call interface	Explanation
Arguments by the stack	Passes function arguments by the stack. This is a default setting.
Arguments by registers	Passes selected function arguments by registers.These processing's reduce the code size and stack usage to improve the program execution speed.

■ Storing Existing Projects in Workspace

1. Open workspace to store a project.
   Open workspace to store a project.
2. Select [Project]-[Add Project]-[Project].
3. Select the project to be stored.
   Select the project to be stored in the [Add Project] dialog.
4. Set project dependence.
   When defining a project as a subproject in another project, place a checkmark in the [Dependence] checkbox and select the project name from the list in the [Dependence] checkbox.
5. Click the [Open] button.

■ Setting Active Project

The active project is a project that undergoes [Make], [Build], [Compile/Assemble], [Start Debug] and [Include Dependence] in the menu. [Make], [Build], [Compile/Assemble], and [Include Dependence] affects the subprojects in the active project.

To set an active project, select [Project]-[Set Active Project]. When the submenu is displayed, select the name of a project to be made active from the submenu.

■ Deleting Project Stored in Workspace

Select the project to be deleted in the SRC tab of Project window. Select [Delete Project] in the shortcut menu.

The specified project is deleted from workspace, but the project file itself is not deleted.

If the deleted project is used as the subproject in the project within workspace, the project dependence is also deleted.

Note:

If SOFTUNE REALOS is not installed, REALOS is not displayed in the project type.
Projects created by V30L31 or earlier version use the function-call interface for "stack argument passing".
For details of the function-call interface, see "Appendix H Guide to Changing Function-call Interface".

2.5 Creating and Registering Source File in Project

This section explains the procedure for creating a new source file with SOFTUNE WORKBENCH and registering the file in the project.

■ Creating the new source file

1. Select the [File] [New] Menu.
   When the [New] dialog box opens, select [Text File] from [Types of Files], then click the [OK] button.
2. Select the [File] [Save As] Menu.
   When the [Save As] dialog box opens, select [Text File] from [Types of Files], then click the [OK] button.
   When the file dialog box for specifying the directory to which the created file is to be saved and the file name opens, select the directory, specify the file name, then click the [Save] button.

■ Registering the created file in the project

Select the [Project]-[Add Member]-[File] menu. The File dialog is opened to select the file to be added to the member. Select the created source file, followed by the folder inserted into the SRC tab of project window, and click the [Open] button. The file is stored in the project and its name is displayed in the specified folder in the SRC tab of Project window.

■ Storing Created File with Directory in Project

Select the [Project]-[Add Member]-[Directory] menu. The Add Member Directory dialog is opened to select the folder to be added to the member. Select the directory having the created source file, followed by the folder to be inserted into the SRC tab of Project window, and click the [OK] button. The file and directory are stored in the project and the file and directory below the specified directory are displayed in the specified folder in the SRC tab of Project window.

The type of file to be stored can be restricted by setting [Type of File] in the dialog.

■ To delete files stored in the project

Select a file(s) to be deleted in the SRC tab of Project window. Select "Delete" from the shortcut menu. The selected file is deleted from the project member, but the file itself is not deleted. Users cannot delete files in the [Dependencies] category and [Debug] category files.

2.6 Definition of Subproject

This section explains how to define a subproject.

■ Definition of Subproject

The subproject is a project on which other projects depend.

SOFTUNE WORKBENCH uses the following methods to define a subproject.

- Defining project as subproject in storing it

  When created, a new project is defined as a subproject in another project. For the setting method, see Section "2.4 Storing Project".

- Defining subproject between existing projects

  A subproject is defined between projects in workspace.

  Another project is defined as a subproject in the subproject in the parent project. Such a recurrent definition that the parent project itself serves as a subproject is impossible.

■ Defining Subproject between Existing Projects

1. Select [Project]-[Project Dependence].
2. Select the parent project in which a subproject is defined.
   When the [Project Dependence] dialog is opened, select the name of the parent project in which a subproject is defined from the [Project Name] box.
3. Select the project that is defined as a subproject.
   Check the project that is defined as a subproject from those in the [Dependent Project] list.
4. Click the [OK] button.
   Select the [Project] [Add Member] menu. The File dialog is opened to select the file to be added to the member. Select the created source file, and click the [Open] button. The file is stored in the project and its name is displayed in the source file category in the Project window.

■ Deleting Subproject from Project

1. Select [Project]-[Project Dependence].
2. Select the parent project from which a subproject is deleted.
   When the [Project Dependence] dialog is opened, select the name of the parent project from which a subproject is deleted form the [Project Name] box.
3. Select the subproject to be deleted.
   Deselect the subproject that is deleted from the [Dependent Project] list.
4. Click the [OK] button.

2.7 Creation of Project Configuration

This section explains how to create a project configuration.

■ Creation of Project Configuration

The project configuration is a series of settings for specifying the characteristics of the target file. By creating a new project configuration, two or more tool settings can be stored in the project.

When a new project is created, the project configuration is created under a default name of "Debug".

In SOFTUNE WORKBENCH, the project configuration is created as follow.

- Creating project configuration on settings of existing project configuration

  A new project configuration is created on the settings of the selected existing project configuration. In the new project configuration, the same files as those in the original project configuration are always used.

■ Creating Project Configuration on Settings of Existing Project Configuration

1. Select [Project]-[Project Configuration]-[Add and Delete].
2. Select the project to which a project configuration is added.
   When the [Add and Delete Project] dialog is opened, select the project to which a project configuration is added.
3. Click the [Add] button.
   Click the [Add] button. The [Add Project Configuration] dialog is opened.
4. Enter the project configuration name.
   Enter the unique name of a new project configuration. The characters that can be used to form a name are "a to z", "A to Z", "0 to 9" and "_".
5. Select the project configuration to which settings are copied.
   Select the initial settings of a project configuration to be added. The selected settings of the project configuration (such as tool options, file configurations, and configurations of subprojects to be build) are copied as they are.
6. Click the [OK] button.
   Click the [OK] button in the [Add Project Configuration] dialog and the [OK] button in the [Add and Delete Project Configuration] dialog.

■ Setting Active Project Configuration

The active project configuration is at default a project configuration that undergoes [Make], [Build], [Compile/Assemble], [Start Debug], and [Include Dependence].

1. Select [Project]-[Project Configuration]-[Add and Delete].
2. Select the project configuration that is made active.
   When the [Add and Delete Project] dialog is opened, select the name of the project configuration that is made active.
3. Click the [Active] button.
   Click the [Active] button. The specified configuration and its project become active.
4. Click the [OK] button.

■ Deleting Specific Active Project Configuration from Project

1. Select [Project]-[Project Configuration]-[Add and Delete].
2. Select the project configuration that is deleted from the project.
   When the [Add and Delete Project] dialog is opened, select the project configuration name to be deleted.
3. Click the [Delete] button.
   Click the [Delete] button. The specified project configuration is deleted. When all project configurations in a project are deleted, the project itself is also deleted.
4. Click the [OK] button.

2.8 Setting Tools

When make or build is executed by SOFTUNE WORKBENCH, appropriate options must be set in such tools as a compiler, assembler and linker. Set these options as follows:

■ Select the [Project] [Setup Project] Menu

The [Set Project] dialog is opened. The option selected in the [Set Project] dialog box is applicable to two or more projects. The applicable project configuration can be limited. For example, the settings of project configurations A and B can be changed. The same setting can be also specified for all project configurations.

Specify the project configuration in the [Setting Target] box, the project set in tree view, and select the tool tab.

When the compiler, assembler and linker/librarian are selected, the category can be selected in the top tab of the setting dialog box for each tool. After the drop-down list is opened, select a category. When a category was selected, the contents in the display are changed and the options included in each category can be set.

In most cases, compiler and assembler options need not be set except when output of list file and make or build is executed under specific conditions. Set only linker options. For how to set linker options, see Section "2.9 Setting Linker Options".

■ Click the [OK] button to complete tool setting

When setting all necessary tool options is completed, click the [OK] button. All the set options are registered in the project; they become valid when make or build is executed.

Clicking the [Cancel] button cancels all the set options.

Note:

When the [Update] button is clicked during tool option setting, the previously set options cannot be restored.

2.9 Setting Linker Options

When creating a program with SOFTUNE WORKBENCH, be sure to set a memory map with a linker option.

■ Automatic Setting of Linker Options

In SOFTUNE WORKBENCH, the following linker options are automatically set on the basis of information on the MCU selected when a new project is created;

- Specify the internal ROM/RAM address of the MCU in the memory area option.

  Internal ROM is output under an area name of "_INROMxx" and internal RAM under an area name of "_INRAMxx" (where x is numbered consecutively starting with 01).

- Set the automatic disposition mode to mode 2 (optimum automatic disposition by linker).

  When creating a program in a mode other than the single-chip mode or when customizing the disposition of sections, set the [Disposition/Connection] option as the linker option.

■ Setting of Linker Options

In SOFTUNE WORKBENCH, specifying memory mapping is basic to the disposition of each section in the memory area. Therefore, set a memory area and set each section in the memory area.

■ Setting of memory area

Enter a ROM/RAM area name, start address and end address, select an area attribute (ROM or RAM), then click the [Setup] button. This setting is displayed in the ROM/RAM area list. In the ROM/RAM area, assign a unique ROM/RAM name so that it does not match other area names.

An easy program consists of a ROM area and a RAM area, but in a complicated program, several area may be specified. The number of areas that can be set is not limited; set all the areas necessary to configure the memory map of the program to be developed.

In Auto Disposition (Mode 2), the linker automatically allocates sections unspecified for allocation in a ROM/RAM area.

The linker searches an available ROM/RAM area beginning at the top of the [ROM/RAM Area List]. Click the [Up] button or the [Down] button to change the desired number.

■ Setting of the sections

Selecting the area from the ROM/RAM area list and clicking the [Setup Section] button opens the [Setup Section] dialog box, enabling the sections to be allocated to the selected area. When selecting an area, click the start address of the area.

When the [Setup Section] dialog box opens, specify section names in the order the sections are allocated to the area. Specify section names one by one. When section name specification is completed, click the [Setup] button to register the section names in the section name list.

When execute make/built the sections are allocated to the area in the order the section names were registered in the section name list.

When setting the sections to be allocated to one area is completed, click the [OK] button to return the linker option setting dialog box. Also set other areas in the same way.

Reference:

Allocating sections to several areas can be continuously set by changing ROM/RAM area name display in the uppermost part of the [Setup Section] dialog box. The linker option setting dialog box need not be returned each time sections are allocated to an area.

2.10 Make/Build

SOFTUNE WORKBENCH can create a program in two methods: make and build.

■ Make

Compiles or assembles only the modified source file and then links all objects to the library to generate an object program. SOFTUNE WORKBENCH recognizes the dependency of the include files registered in the [Dependencies] category of the SRC tab of Project Window to compile or assemble the source file.

Use the [Project]-[Make] Menu to execute make.

■ Build

Not only modified source file, but compiles or assembles all the source files registered in the project and then links all objects to the library to generate a target file.

Use the [Project]-[Build] Menu to execute build.

■ Stop

Stop is the function that forcibly suspends processing during make, build, compilation, or assembling.

Execute stop with the [Project]-[Stop] Menu during make, build, compilation, or assembling.

2.10.1 Making or Building of Project

SOFTUNE WORKBENCH enables making or building for each project configuration.

■ Making or Building of Project

[Make] or [Build] in the menu applies to the active configuration of an active project. If a subproject is defined, priority is given to making or building of the subproject.

For details about how to change the active project and active configuration, see Section "2.4 Storing Project" and Section "2.7 Creation of Project Configuration".

■ Making or Building Specified Project

Select the project to be made or built in the SRC tab of Project window. Select [Make] or [Build] in the shortcut menu. The active configuration of the specified project is made or built. If a subproject is defined, priority is given to making or building of the subproject.

■ Changing Subproject Configuration at Making or Building

1. Select [Project]-[Project Configuration]-[Configuration at Build].
2. Select the parent project and configuration.
   When the [Set Configuration at Build] dialog is opened, select the project to be set from the [Project] box. The configuration of a subproject to be made or built is displayed.
3. Select the configuration of a subproject.
   Select the configuration to be made or built from [Configuration of Subproject at Make/Build].
4. Click the [OK] button.
   Stop is the function that forcibly suspends processing during make, build, compilation, or assembling. Execute stop with the [Project]-[Stop] Menu during make, build, compilation, or assembling.

2.11 Debugging

The absolute file created as a result of normal termination of make/build can be debugged immediately after SOFTUNE WORKBENCH has migrated to the debug session.

■ Migrating SOFTUNE WORKBENCH to debug session

To enable SOFTUNE WORKBENCH to debug the absolute file, migrate it to the debug session.

To migrate SOFTUNE WORKBENCH to the debug session, select [Start Debug] from the [Debug] Menu

-First debugging after project creation

Setup Wizard for debuggers is started. Set the type of the debugger and others ("4.7.2.5 Setup Wizard").

-Second or subsequent debugging after project creation

Start SOFTUNE WORKBENCH in the debugger mode that has been already set. To change the type of the debugger, select the [Debug]-[End] Menu to terminate debugging once, and then change [Project]-[Project Setup] Menu.

■ Automatic downloading of monitor program

In case the following emulator is used, the debugging environment is determined from the emulator type and version, etc., of the downloaded monitor program when proceeding to the debug session to download the optimum monitor program automatically (Appendix B Downloading Monitor Program).

MB2147-01/MB2147-05

■ Loading the target program

When SOFTUNE WORKBENCH enters the debug session, select [Load target program] from the [Debug] Menu to load the target program. The created program is loaded to the debugger and all debugging preparations are completed now.

■ Operating the debugger

For how to operate the debugger, see Chapter 3 Windows and Chapter 4 Menus in this manual.

For debugger commands and debugger output error messages, refer to the SOFTUNE WORKBENCH Command Reference Manual

For the debugger functions (MCU common function, MCU chip dependency function), refer to the SOFTUNE WORKBENCH Users Manual

2.12 Executing Debugging Only

SOFTUNE WORKBENCH can be used as the conventional debuggers.

■ Creating a project

In SOFTUNE WORKBENCH, projects are a basis of all work. This is not an exception also at debugging. For this reason, executing debugging only requires the creation of a project for debugging.

First, create the project for debugging and the workspace which stores the project in the following procedure:

1. Select [New] from the [File] Menu.
2. Select [Project/Workspace File] from the file open dialog box.

   -Select the absolute format (ABS) from the new creation dialog box for the project.

   -Specify a project name.

   -Specify a project directory.

   -Select a target MCU name and chip type.

■ Setting of Workspace

Perform setting common to projects to be stored in workspace.

1. Select [Set Workspace] from the [Project] menu.
2. When the [Set Workspace] dialog is opened, perform the following settings
   - Debug when workspace opened: Start debugging.
   - Save setup information: Save.

■ Settings related to the debugger

1. Select [Project Setup] from the [Project] Menu.
2. When the setup dialog box opens, open the [Debug] tag and select [Setup] category:
3. Set a setup name.
   A project name is set both in [Setup Name List] and [Valid Setup Name] as the default setup name. Usually, setup names are identified by the type of the debugger to be used. However, if only one debugger is used, the default name may be used as it is.
   When the default name is used select the default name already set in [Setup Name List], then click the [Change Setup] button. When another name is used, specify [Setup Name], then click the [Add Setup] button.

■ Setup Wizard

Clicking the [Add Setup] or [Change Setup] button starts the Setup Wizard for debuggers. Once Setup Wizard has been started, set items according to instructions from Setup Wizard. For how to set items with Setup Wizard, see Section "4.7.2.5 Setup Wizard".

When all settings with Setup Wizard are completed, click the [Complete] button.

When the [Project Setup] dialog box is redisplayed, click the [OK] button.

When all the above steps are completed, save the project, then close it once.

■ Starting debugging

When steps from [Creating a project] to [Setup Wizard] are complete, open the project. SOFTUNE WORKBENCH automatically migrates to the debug session, enabling the immediate start of debugging.

Select [Open] from the [File] menu and specify the load module file to load the target program.

2.13 Reading SOFTUNE Project Files of Old Versions

The SOFTUNE project files of old versions can be read.

■ Procedure

- The project files created in SOFTUNE WORKBENCH V3 version need the following setting.

1. Select [Open Workspace File] from the [File] menu.
2. Select 'Project file' from File Type and specify the project file made by the early version of SOFTUNE WORKBENCH. If the specified file is one made by the early version of Softune Workbench, a dialog asking whether to convert the file to a workspace project format is opened.

[Yes] button: The project file is converted and opened in the workspace project format.

[No] button: The project file is not converted and is opened in the old project format. In this case, some functions cannot be used. For details of functions that can be used in the old project format, refer to Section 1.2 Management Function for Project of SOFTUNE WORKBENCH Users Manual.

[Cancel] button: Opening of the project file is cancelled.

- The project files created in SOFTUNE Manager V01/V02 version needs the following setting.

1. Select [Open Workspace File] from the [File] Menu.
2. Select 'Project file' from File Type and specify the project file made by SOFTUNE Manager. If the specified file is one made by SOFTUNE Manager, a dialog asking whether to convert the file to a workspace project format is opened.
   Click the [Cancel] button to cancel opening of the project file.
3. Click the [OK] button to start conversion.
   When you click the [Cancel] button, it cancels the opening of the project file.
4. When the New Project Creation Window opens, set the chip type and target MCU, then click the [OK] button.
5. When conversion is completed, the dialog box showing the end of conversion opens. Click the [OK] button to close the dialog box.

■ Backup file

In SOFTUNE WORKBENCH, when a project file is converted to a workspace project format, a backup file is made automatically. The extension of backup file varies with the type of project file. The method for opening the backup project file is also different depending on the extension.

● SOFTUNE WORKBENCH V3

Old project file (.prj) → .p03

Old option data file (.dat) → .d03

● SOFTUNE WORKBENCH V01/V02

Old project file (.prj) → .V01

● SOFTUNE WORKBENCH V3 (.p03)

- Change the extension of the backup project file (.p03) and option data file (.d03) to '.prj' and '.dat,' respectively.

● Softune Manager V01/V02 (.v01)

- Change the extension of the backup project file (.v01) to '.prj.'

Note:

- Tool options are not passed to projects of SOFTUNE Manager V01/V02. Reset these options after read has terminated.
- Be sure to reset "User Include File Directory" set by "Set Environment Variable" of an old version as the "Include Path" option of the C compiler or assembler. Also be sure to reset "Library File Search Path" set by "Environment Variable Setup" of the old version as the "Library Search Path" option of the linker.
- If the workspace file having the same name as that of the specified project file is in the same directory, the workspace file is opened instead of the project file, and no project file is converted.

2.14 Moving Project Files

This section explains how to move a project file to another directory or a personal computer.

■ Procedure

1. Set the path to the member stored in the project file to the relative path from the project file.
   In SOFTUNE WORKBENCH, files in the same drive as that of the project file are usually stored in the relative path. To check whether the files are stored in the relative path, check File Property in the SRC tab of Project window. For the file property, see Section "4.3.9 Property"
2. Set the path to the target file directory, object file directory and list file directory to the relative path from the project file.
   In SOFTUNE WORKBENCH, when a new project is created, the output directory is set to the relative path from the project file. To make a change and check, open the [Set Project] dialog. For details, see Section "4.5.5 Setup Project".
3. Set the directories such as the include path and library path specified in the tool option to the relative path or macro description.
   For the macro description, see Section "1.11 Macro Descriptions Usable in Manager" of SOFTUNE WORKBENCH Users Manual. For the tool option, see Section "4.5.5 Setup Project".
4. After the completion of the setting in steps 1 to 3, save the project.
5. Move the project file without changing the directory structure in steps 1 to 3.

2.15 Useful Functions

This section explains some useful functions to use in the SOFTUNE Workbench.

■ Tab Display of the Windows

In the SOFTUNE Workbench, the displayed windows are tabbed.

This makes it easier to find the desired window quickly when multiple windows are displayed.

■ Drag & Drop between Windows

SOFTUNE Workbench supports the drag & drop function between windows in the debugger. The followings are the windows that currently support the drag & drop function.

Please refer to the description of each window for the operation by the drag & drop.

- Project window
- Source window
- Memory window
- Disassemble window
- Watch window
- Coverage window
- Real-time memory window
- RAM checker window

■ Confirmation of the Setting

SOFTUNE Workbench has various dialogs to set up functions. Each dialog requires you to click Set button to take effect after you set up. If you attempt to close the dialog without clicking the [Set] button, a dialog for confirmation will be displayed so that you won’t fail to set up.

CHAPTER 3
Windows

This chapter explains SOFTUNE WORKBENCH windows.

3.1 Window Configuration
3.2 Tool Bar
3.3 Status Bar
3.4 Project Window
  3.4.1 SRC tab
  3.4.2 ABS tab
3.5 Output Window
3.6 Edit Window (Standard Editor)
  3.6.1 Setting Standard Editor
3.7 Source Window
3.8 Symbol Window
3.9 Disassemble Window
3.10 Register Window
  3.10.1 Setting Register Display
3.11 Memory Window
3.12 Local Variable Window
3.13 Watch Window
3.14 Trace Window
3.15 Command Window
  3.15.1 Setting Character String Replacement
  3.15.2 Logging
3.16 Object Window
3.17 Coverage Window
3.18 Performance Window

3.19 Sequence Window
3.20 Real-time Memory Window
3.21 RAM Checker Window
3.21.1 Setup of the RAM Checker
3.21.2 Start the RAM Checker viewer

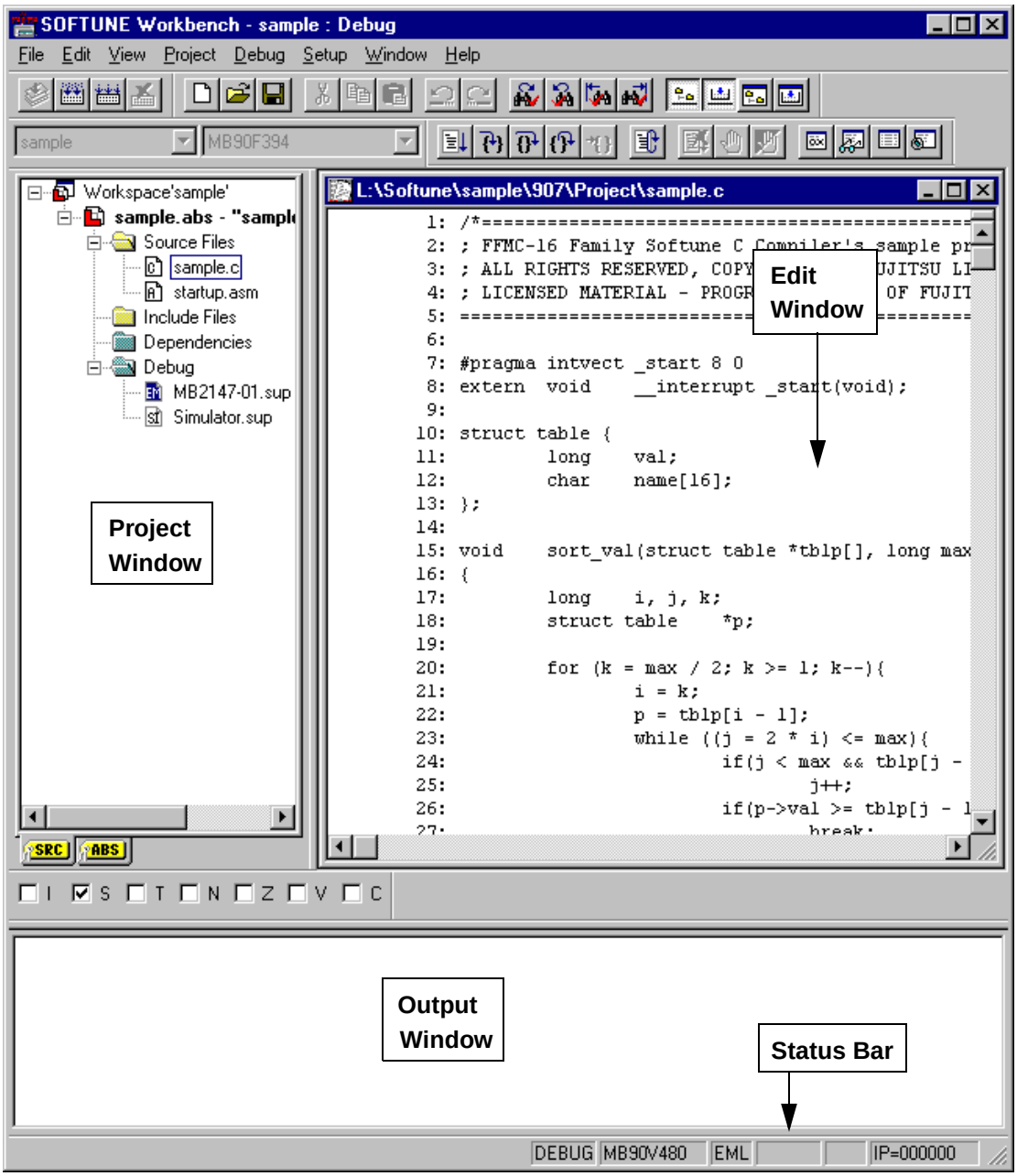
3.1 Window Configuration

Each SOFTUNE WORKBENCH window consists of a menu bar, tool bar, window screen, and status bar. Menus are explained in Chapter 4.

■ Main Window

Figure 3.1-1 shows the SOFTUNE WORKBENCH Main Window. As shown in this figure, child windows (e.g., project window and output window) and the tool bar can be shown with the Main Window and displayed.

Figure 3.1-1 The SOFTUNE WORKBENCH Main Menu



3.2 Tool Bar

Command buttons to be used often are assigned to the tool bar for each group. The groups that can be selected and the command buttons in the groups are shown below.

■ Common bar

New

Open

Save

Cut

Copy

Paste

Undo

Redo

Jump to Next error

Jump to Previous error

Jump to Top error

Jump to Bottom error

Project window Docking

Output window Docking

Open Project window

Open Output window

■ Project bar

The active project and its active configuration name are set and displayed.

■ Build bar

Compile/Assembly

Make

Build

Stop Make/Build

■ Debug bar

Go

Step In

Step Over

Step Out

Run Until Cursor

Reset MCU

Abort

Breakpoint Set/Reset

Register window

Watch window

Memory window

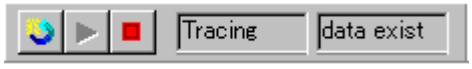
Disassembly

■ Flag bar

Display and setting of MCU condition flag status

■ Trace

Display of the trace acquisition state and trace control while the user program is running



Update of the trace window

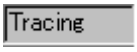
Updates the trace window. When the trace data is being acquired, the window is updated after a forced stop.

Forced start of the trace acquisition

Forces to resume the acquisition of the trace data. It becomes valid only when the trace acquisition has been forced to stop. It is only valid while running MCU. Forced stop of the trace acquisition

Forced stop of the trace acquisition

Forces to stop the acquisition of the trace data. This doesn't stop any running user program, except the trace acquisition. It is only valid while running MCU.



Trace acquisition state

Indicates the current trace acquisition state.

Tracing : Trace data is being acquired.

Pause : Trace acquisition is being paused.

End : Trace acquisition has been terminated.

Trace buffer state

Indicates the current trace buffer state.

Data exist : Data exist in the trace buffer.

no data : No data exist in the trace buffer.

Buffer full : Trace buffer is full.

Note:

Trace tool bar becomes valid only when the debug function of MB2147-01 is in Trace Enhancement mode. For more information, please refer to Section "2.3.1.6 Debug Function" in SOFTUNE Workbench User's Manual.

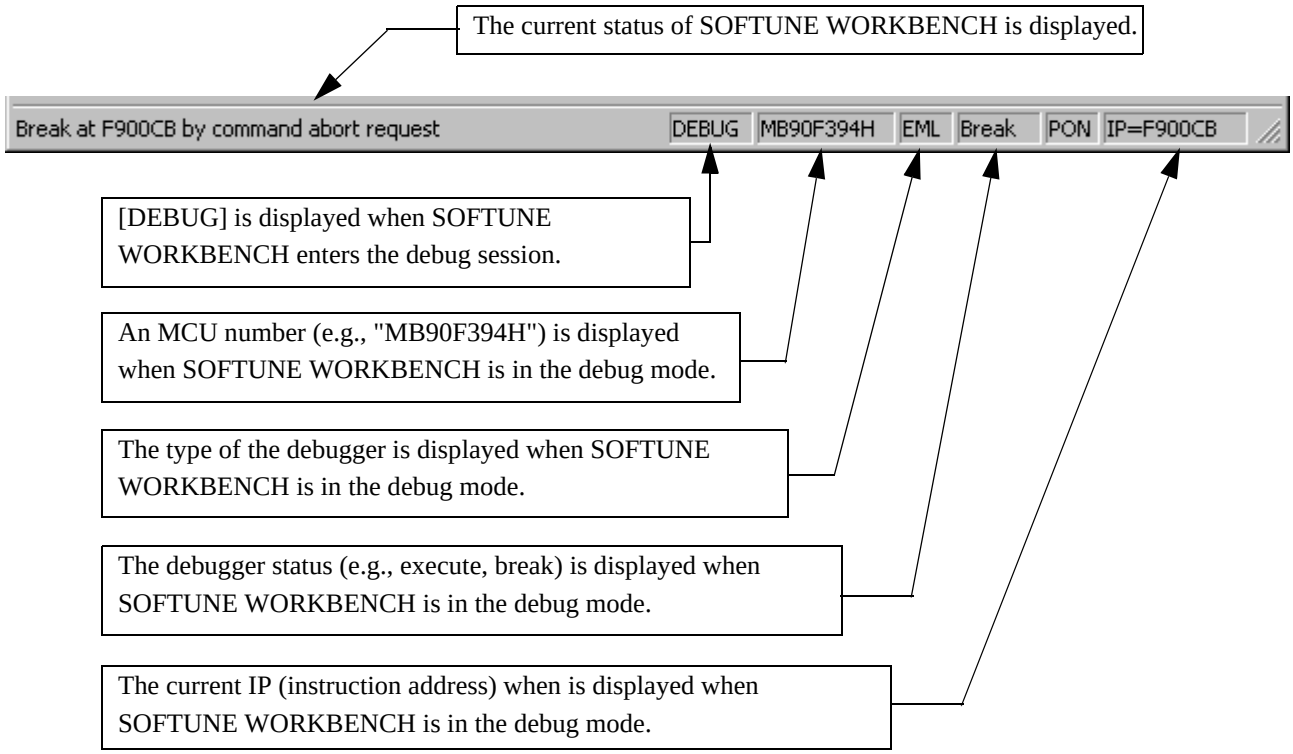
3.3 Status Bar

The current status of SOFTUNE Workbench is displayed. The status bar is displayed in the last part of the main window.

■ Status bar

The status bar has the areas in which information is displayed. The information and the areas in which it is displayed are simply shown in Figure 3.3-1.

Figure 3.3-1 Status Bar


The current status of SOFTUNE WORKBENCH is displayed.
Break at F900CB by command abort request
DEBUG
MB90F394H
EML
Break
PON
IP=F900CB
[DEBUG] is displayed when SOFTUNE WORKBENCH enters the debug session.
An MCU number (e.g., "MB90F394H") is displayed when SOFTUNE WORKBENCH is in the debug mode.
The type of the debugger is displayed when SOFTUNE WORKBENCH is in the debug mode.
The debugger status (e.g., execute, break) is displayed when SOFTUNE WORKBENCH is in the debug mode.
The current IP (instruction address) when is displayed when SOFTUNE WORKBENCH is in the debug mode.

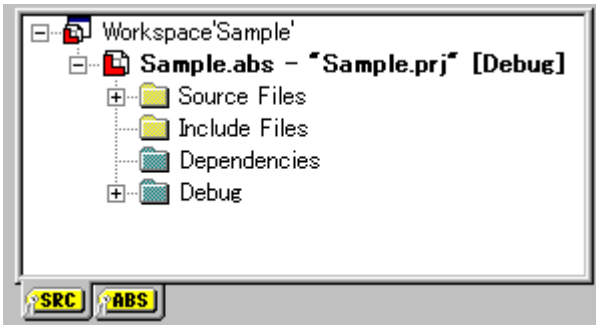
3.4 Project Window

The Project Window displays information about the project and load module.

■ Project window

Figure 3.4-1 shows an example of the project window.

Figure 3.4-1 Project Window



To select the window display contents, use the tab on the bottom of the project window. The project window has the following tabs.

- SRC tab

  Displays information on the project. Refer to Section "3.4.1 SRC tab" tab for details.

- ABS tab

  Displays information on the source file acquired from the debug information. Refer to Section "3.4.2 ABS tab" for details.

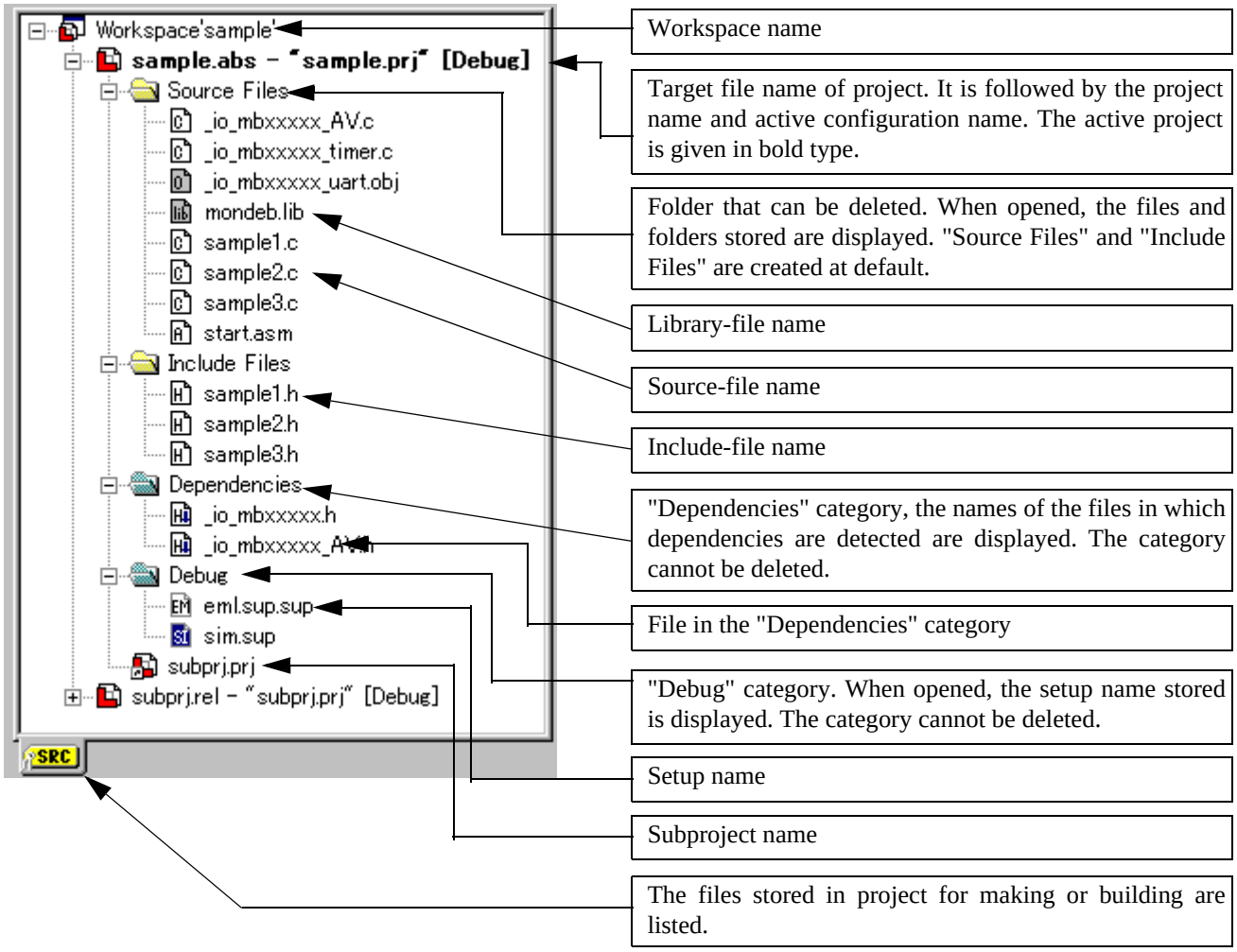
3.4.1 SRC tab

The name of the current project and the file names registered in the project are displayed in the tree view format.

■ SRC tab

Figure 3.4-2 shows examples of displayed contents of the SRC tab.

Figure 3.4-2 SRC tab


Workspace'sample'
sample.abs - "sample.prj" [Debug]
Source Files
_io_mbxxxxx_AV.c
_io_mbxxxxx_timer.c
_io_mbxxxxx_uart.obj
mondeb.lib
sample1.c
sample2.c
sample3.c
start.asm
Include Files
sample1.h
sample2.h
sample3.h
Dependencies
_io_mbxxxxx.h
Debug
eml.sup.sup
sim.sup
subprj.prj
subprj.rel - "subprj.prj" [Debug]
SRC
Workspace name
Target file name of project. It is followed by the project name and active configuration name. The active project is given in bold type.
Folder that can be deleted. When opened, the files and folders stored are displayed. "Source Files" and "Include Files" are created at default.
Library-file name
Source-file name
Include-file name
"Dependencies" category, the names of the files in which dependencies are detected are displayed. The category cannot be deleted.
File in the "Dependencies" category
"Debug" category. When opened, the setup name stored is displayed. The category cannot be deleted.
Setup name
Subproject name
The files stored in project for making or building are listed.

■ Function

- Display the workspace name
- Display all projects stored in workspace
- Display the subproject

  The subproject in the project is displayed below the parent project.
- Display of the project target file name
- Display of all the source file names registered in the project

  When a source file name is double-clicked, the editor opens to enable the editing of the source file.
- Display of the include files that are in dependency

  When an include file name is double-clicked, the editor opens to enable the editing of the include file.
- Displaying Debugger setup name

  When the Debugger setup name is double-clicked, debugging is started based on the setup information.
- Drug and drop

  The member in the SRC tab can be moved in the Project window and stored from the explorer.

■ SRC tab bitmap image list

Workspace file name

Target file name

Subproject name

Folder

Category

C source file

C source file (not subject to making or building)

C source file in which individual options specified

C source file in which individual options specified (not subject to making or building)

Assembler source file

Assembler source file (not subject to making or building)

Assembler source file in which individual options specified

Assembler source file in which individual options specified (not subject to making or building)

Include file

Include file (with dependence)

Library file

Library file (not subject to making or building)

Object file

Object file (not subject to making or building)

Relative format file

Relative format file (not subject to making or building)

REALOS configuration file

Other user's registration files

Simulator Debugger setup name

Valid setup name (for Simulator Debugger)

Emulator Debugger setup name

Valid setup name (for Emulator Debugger)

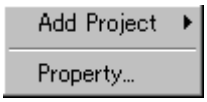
Monitor Debugger setup name

Valid setup name (for Monitor Debugger)

■ Shortcut menus (Click the right button of the mouse on a workspace name)

Figure 3.4-3 shows a shortcut menu.

Figure 3.4-3 shortcut menu on a workspace name



- Add Project

  When selected, the following two submenus are displayed.

  - New

    A new project is added to workspace (See Section "4.5.2 Add Project").

  - Existing Project

    An existing project is added to workspace (See Section "4.5.2 Add Project").

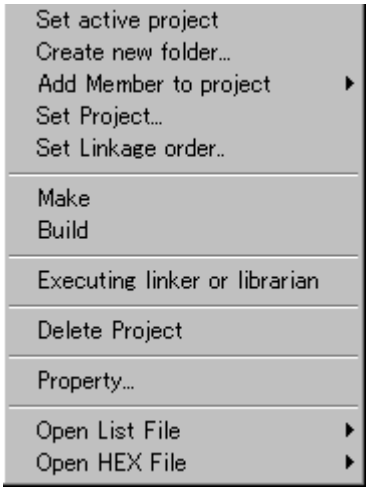
- Property

  Information on the workspace file is displayed (See Section "4.3.9 Property").

■ Shortcut menus (Click the right button of the mouse on a target name)

Figure 3.4-4 shows a shortcut menu.

Figure 3.4-4 shortcut menu on a target name



- Set Active Project

  The currently selected project is set in the active project in workspace.

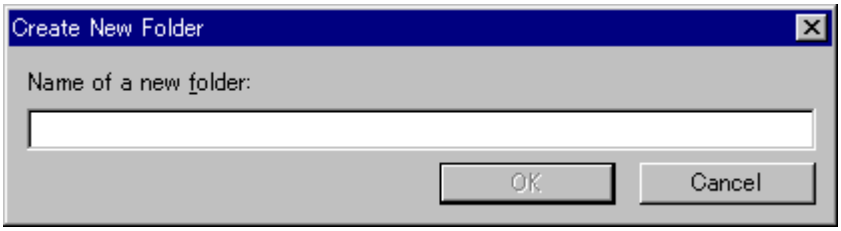
- Create New Folder

  When selected, the [Create New Folder] dialog (Figure 3.4-5) is opened. Enter the folder name and click the [OK] button.

  When the folder name is entered, the folder is inserted into the hierarchy immediately below the project.

  The folders are listed in the order in which they are created.

Figure 3.4-5 Create New Folder Dialog



- Add Member to Project

  When selected, the following two submenus are displayed.

  - File

    A member is added in files to the currently selected project (See Section "4.5.3 Add Member").

    At default, a member is inserted into the hierarchy immediately below the project.

  - Directory

    A member is added in directories to the currently selected project (See Section "4.5.3 Add Member").

    At default, a member is inserted into the hierarchy immediately below the project.

- Set Project

  A project is set (See Section "4.5.5 Setup Project").

- Set Linkage Order

  When selected, the [Set linkage Order] dialog (Figure 3.4-6) is opened. The files displayed in the [Link Order] box are linked from top to down. To change the link order, select the configuration to be set from the [Setting Target] box, followed by the file name, and use the [Up] or [Down] button to move the file to a desired position.

  The [Export] dialog (Figure 3.4-7) allows the current order to affect other configurations.

  The files not subject to link are displayed in gray.

Figure 3.4-6 Set Linkage order Dialog

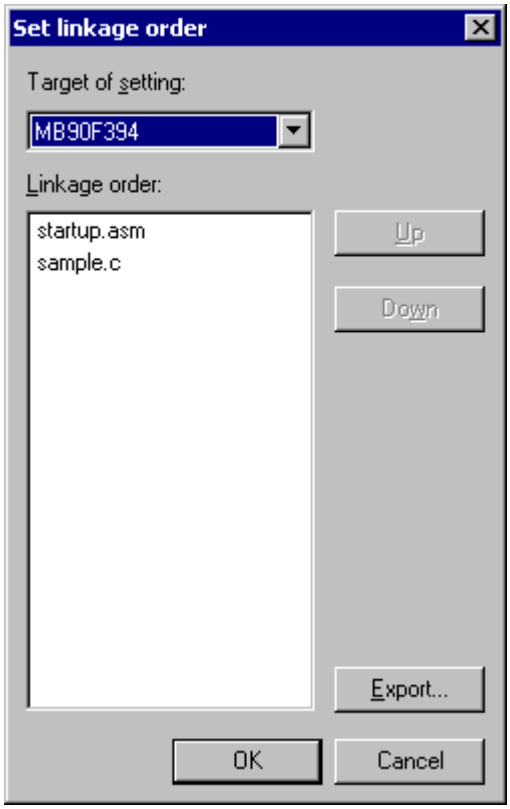
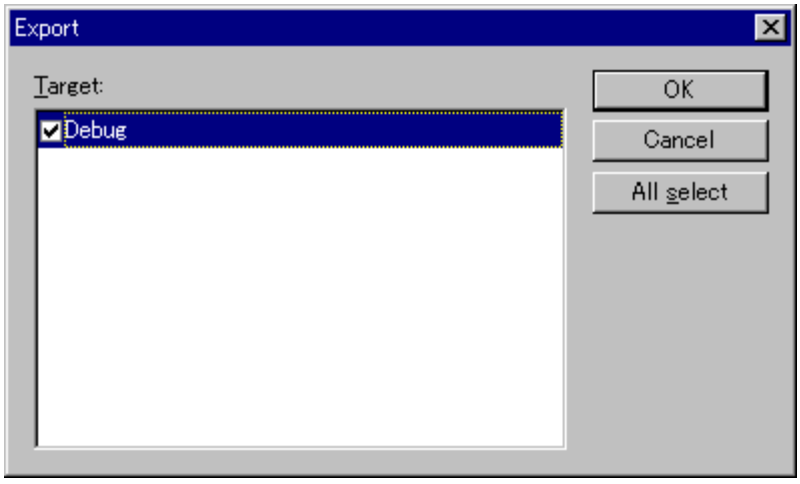


Figure 3.4-7 Export Dialog



- Make/Build

  Making or building is performed in the active configuration of the currently selected project to create a target file.

- Executing linker or librarian

  The linker or librarian is started in the active configuration of the currently selected project to create a target file.

- Delete Project

  The specified project is deleted from workspace. The project file itself is not deleted.

- Property

  File information is displayed. (See Section "4.3.9 Property")

- Open List File

  Select the list file you want to open, from the submenu.

  If the list file you want to open is not created, no submenu item can be selected.

- Open HEX File

  Select the HEX file you want to open, from the submenu.

  If the HEX file you want to open is not created, no submenu item can be selected.

■ Shortcut menu (Click the right button of the mouse on a subproject name)

Figure 3.4-8 shows a shortcut menu.

Figure 3.4-8 shortcut menu on a subproject name

- Delete Subproject

  The subproject in the specified project is deleted. To define a subproject again, see Section "4.5.7 Project Dependencies".

■ Shortcut menus (Click the right button of the mouse on a folder)

Figure 3.4-9 shows a shortcut menu.

Figure 3.4-9 shortcut menu on a folder

- Create New Folder

  When selected, the [Create New Folder] dialog (Figure 3.4-5) is opened. Enter the folder name and click the [OK] button.

  When the folder name is entered, a folder is inserted into the selected folder.

  The folders are listed in the order in which they are created.

- Add Member to Folder

  When selected, the following two submenus are displayed:

  - File

    A member is added in files to the currently selected project (See Section "4.5.3 Add Member").

    At default, a member is inserted into the selected folder.

- Directory

  A member is added in directories to the currently selected project (See Section "4.5.3 Add Member").

  At default, a member is inserted into the selected folder.

- Set

  When selected, the following two submenus are displayed:

  - Individual Setting

    A project is set (See Section "4.5.5 Setup Project").

  - Return to Common Setting

    All individual options in the specified folder are returned to common options.

- Make/Build

  Making or building is performed in the active configuration of the currently selected project to create a target file.

- Delete

  The selected folder and all files in the folder are deleted from the project. The files themselves are not deleted.

  If the files (RCF files) that cannot be deleted are included, the folder is not deleted.

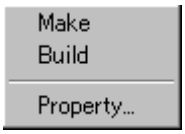
- Property

  Information on folders is displayed (see Section "4.3.9 Property").

■ Shortcut menus (Click the right button of the mouse on a "Dependencies" category)

Figure 3.4-10 shows a shortcut menu.

Figure 3.4-10 shortcut menu on a "Dependencies" category



- Make/Build

  Making or building is performed in the active configuration of the currently selected project to create a target file.

- Property

  Information on folders is displayed (see Section "4.3.9 Property").

■ Shortcut menus (Click the right button of the mouse on a "Debug" category)

Figure 3.4-11 shows a shortcut menu.

Figure 3.4-11 shortcut menu on a "Debug" category

- Add Setup

  When selected, the following two submenus are displayed:

  - New

    When selected, the [Create New Setup] dialog (Figure 3.4-5) is opened. Enter the setup name and click the [OK] button.

    When the setup name is entered, the setup wizard is started. For details about the setup wizard, See Section "4.7.2.5 Setup Wizard".

  - Reference

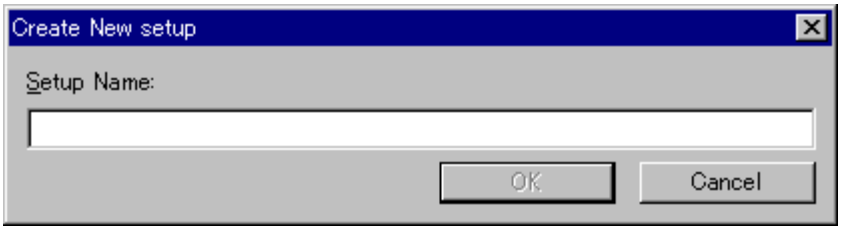
    When selected, the [Create New Setup] dialog (Figure 3.4-12) is opened. Enter the setup name and click the [OK] button.

    When the setup name is entered, the [Select File] dialog is opened. Select the file and click [Open] button. Information is read from the specified file.

- Property

  Information on folders is displayed (see Section "4.3.9 Property").

Figure 3.4-12 Create New setup



■ Shortcut menus (Click the right button of the mouse on a source file name)

Figure 3.4-13 shows a shortcut menu.

Figure 3.4-13 shortcut menu on a source file name

- Open

  When SOFTUNE WORKBENCH is in the debug session, the specified source file is displayed in the Source Window.

  When SOFTUNE WORKBENCH is not in the debug session, the specified source file is displayed in the edit window.

- Compile (Assemble)

  The specified source file is automatically judged whether it is the C or assembler source and then compiled or assembled.

- Set

  When selected, the following two submenus are displayed:

  - Individual Setting

    A project is set (See Section "4.5.5 Setup Project").

  - Return to Common Setting

    All individual options in the specified folder are returned to common options.

- Delete

  The specified file is released from the project member. The file itself is not deleted.

- Property

  File information is displayed. (See Section "4.3.9 Property").

- Open List File

  Select the list file you want to open, from the submenu.

  If the list file you want to open is not created, no submenu item can be selected.

- Open Assemble file

  This menu is opened when a C source file is selected.

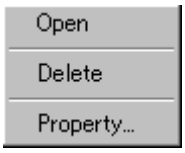
  Of the specified C source file, the assembler source file output by a compiler is opened.

  No file can be selected when not compiled.

■ Shortcut menus (Click the right button of the mouse on an include file and other user's registration file)

Figure 3.4-14 shows a shortcut menu.

Figure 3.4-14 shortcut menu on a include file name



- Open

  When SOFTUNE WORKBENCH is in the debug session, the specified include file is displayed in the Source Window.

  When SOFTUNE WORKBENCH is not in the debug session, the specified include file is displayed in the edit window.

- Delete

  The specified file is released from the project member. The file itself is not deleted.

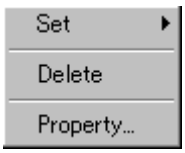
- Property

  File information is displayed. (See Section "4.3.9 Property").

■ Shortcut menus (Click the right button of the mouse on library file, object file and relative format file)

Figure 3.4-15 shows a shortcut menu.

Figure 3.4-15 shortcut menu on a library file name



- Set

  When selected, the following two submenus are displayed:

  - Individual Setting

    A project is set (See Section "4.5.5 Setup Project").

  - Return to Common Setting

    All individual options in the specified folder are returned to common options.

- Delete

  The specified file is released from the project member. The file itself is not deleted.

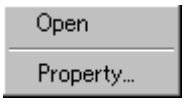
- Property

  File information is displayed. (See Section "4.3.9 Property").

■ Shortcut menus (Click the right button of the mouse on a file in the "Dependencies" category)

Figure 3.4-16 shows a shortcut menu.

Figure 3.4-16 shortcut menu on a file in the "Dependencies" category



- Open

  When SOFTUNE WORKBENCH is in the debug session, the specified source file is displayed in the Source Window.

  When SOFTUNE WORKBENCH is not in the debug session, the specified source file is displayed in the edit window.

- Property

  File information is displayed. (See Section "4.3.9 Property").

■ Shortcut menu (Click the right button of the mouse on setup name)

Figure 3.4-17 shows a shortcut menu.

Figure 3.4-17 shortcut menu on a setup name

- Start /End Debug

  During debugging session, the confirmation dialog "Is the current session ended to start debugging?" is displayed. Select Yes.

  The current debugging is ended, and then debugging is started based on the specified setup information.

  During non-debugging session, debugging is started without displaying the confirmation dialog.

  Debugging can be started only in the active project.

  In debug session, only the valid setup changes from [Start Debug] to [End Debug]. When [End Debug] is selected, debugging is ended.

- Change

  The setup wizard for changing the setup setting is started (see Section "4.7.2.5 Setup Wizard").

- Delete

  Debugger setup information is deleted from the project.

  Valid setup information cannot be deleted. To delete, change the valid setup temporarily.

- Property

  Setup information is displayed (see Section "4.3.9 Property").

■ Shortcut menus (Click the right button of the mouse on a space in the project window)

Figure 3.4-18 shows a shortcut menu.

Figure 3.4-18 shortcut menu on space in the project window

- Docking View

  The project window is docked with the frame in the check status.

- Hide

  The project window enters the nondisplay status.

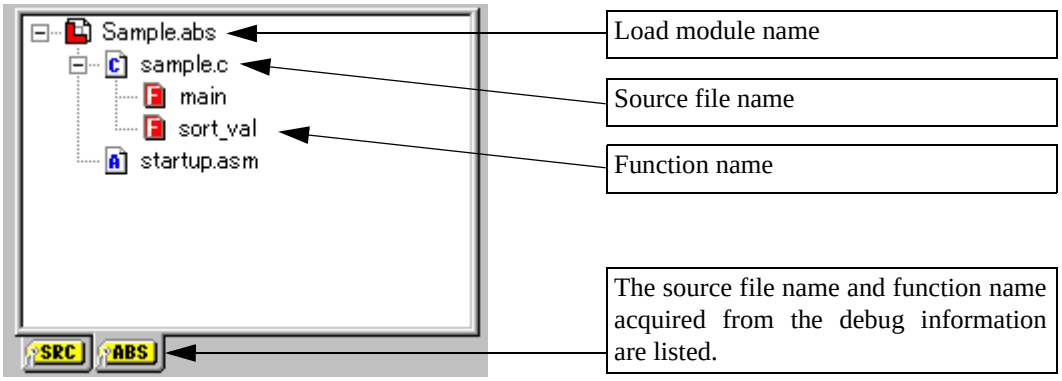
3.4.2 ABS tab

The name of the current project and the file names registered in the project are displayed in the tree view format.

■ ABS tab

Figure 3.4-19 shows examples of displayed contents of the ABS tab.

Figure 3.4-19 ABS tab



■ Function

Display the following information form the debug information.

- Load module name
- Source filename

  Double-click the source file name to open the source window.
- Function name

  Double-click the function name to jump to the source window of the function definition position.
- Coverage rate

  Indicates the coverage rates of load module, source file, and function. Update of the ABS tab is required to display the latest coverage rate.

  The coverage rate is displayed as "(--%)" when all areas are out of range of the coverage measurement. If a part of the area falls in the outside of the coverage measurement rage, an asterisk "*" is attached to the end of the coverage rate.

  This function only becomes valid while MB2141/MB2147-01 emulator debugging.

■ ABS tab bitmap image list

Load module name

C source file

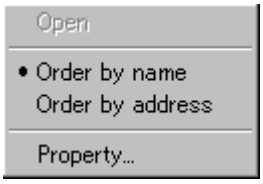
Assembler source file

Function name

■ Shortcut menus (Click the right button of the mouse on a load module name)

Figure 3.4-20 shows a shortcut menu.

Figure 3.4-20 shortcut menu on a load module name



- Open

  Cannot be selected.

- Order by name

  Sort items in the tree in alphabetical order.

- Order by address

  Sort functions in the tree in address order, and sort other items in the tree in alphabetical order.

- Order by coverage rate

  Sort items in the tree in ascending order.

  It is only valid while MB2141/MB2147-01 emulator debugging.

- Set coverage

  Opens the coverage setting dialog.

  It is only valid while MB2141/MB2147-01 emulator debugging.

- Display coverage rate

  Switches on/off the display of the coverage rate. When Display coverage rate is turned on, the coverage rate of each item is displayed. It is only valid while MB2141/MB2147-01 emulator debugging.

- Refresh

  Updates the contents of the window. When Display coverage rate is turned on, the display is updated with the latest coverage rate. It is only valid while MB2141/MB2147-01 emulator debugging.

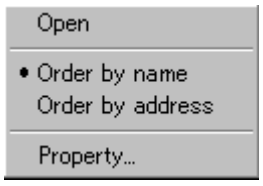
- Property

  Information on the load module file is displayed (See Section "4.3.9 Property").

■ Shortcut menus (Click the right button of the mouse on a source file name)

Figure 3.4-21 shows a shortcut menu.

Figure 3.4-21 shortcut menu on a source file name



- Open

Opens the selected source file in the source window.

- Order by name

  Sort items in the tree in alphabetical order.

- Order by address

  Sort functions in the tree in address order, and sort other items in the tree in alphabetical order.

- Order by coverage rate

  Sort items in the tree in ascending order.

  It is only valid while MB2141/MB2147-01 emulator debugging.

- Set coverage

  Opens the coverage setting dialog.

  It is only valid while MB2141/MB2147-01 emulator debugging.

- Display coverage rate

  Switches on/off the display of the coverage rate. When Display coverage rate is turned on, the coverage rate of each item is displayed. It is only valid while MB2141/MB2147-01 emulator debugging.

- Refresh

  Updates the contents of the window. When Display coverage rate is turned on, the display is updated with the latest coverage rate. It is only valid while MB2141/MB2147-01 emulator debugging.

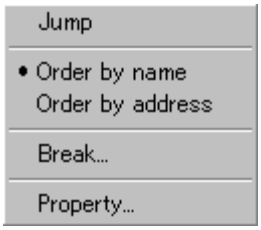
- Property

  Display the properties of the source file.

■ Shortcut menus (Click the right button of the mouse on a function name)

Figure 3.4-22 shows a shortcut menu.

Figure 3.4-22 shortcut menu on a function name



- Jump

  Opens the selected function in the source window.

- Order by name

  Sort items in the tree in alphabetical order.

- Order by address

  Sort functions in the tree in address order.

- Order by coverage rate

  Sort items in the tree in ascending order.

  It is only valid while MB2141/MB2147-01 emulator debugging.

- Break

  Opens a break dialog box to set a break point at the selected function.

- Set coverage

  Opens the coverage setting dialog.

  It is only valid while MB2141/MB2147-01 emulator debugging.

- Display coverage rate

  Switches on/off the display of the coverage rate. When Display coverage rate is turned on, the coverage rate of each item is displayed. It is only valid while MB2141/MB2147-01 emulator debugging.

- Refresh

  Updates the contents of the window. When Display coverage rate is turned on, the display is updated with the latest coverage rate. It is only valid while MB2141/MB2147-01 emulator debugging.

- Property

  Display the properties of the symbol information of the selected function.

■ Shortcut menu (Click the right button of the mouse on space in the project window)

Figure 3.4-23 shows a shortcut menu.

Figure 3.4-23 shortcut menu on space in the project window

- Docking View

  The project window is docked with the frame in the check status.

- Hide

  The project window enters the nondisplay status.

3.5 Output Window

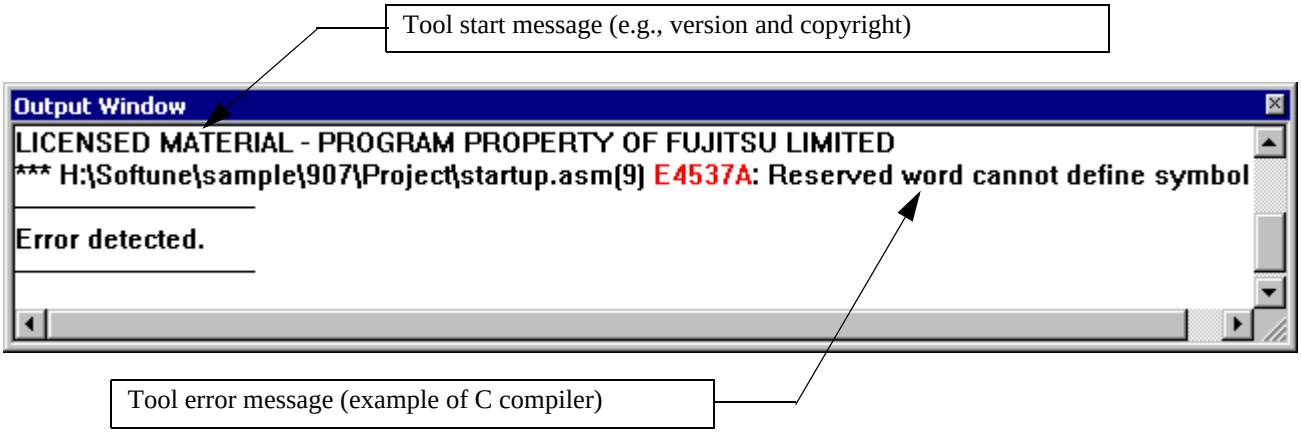
The output window displays compiler version information and error messages, etc., when make or build is executed.

■ Output window

Figure 3.5-1 is an example of the output window.

As shown in this example, when the left button of the mouse is double-clicked on the line to which an error message was output, control automatically jumps to the source line where the error occurred.

Figure 3.5-1 Example of Output Window


Tool start message (e.g., version and copyright)
Output Window
LICENSED MATERIAL - PROGRAM PROPERTY OF FUJITSU LIMITED
*** H:\Softune\sample\907\Project\startup.asm(9) E4537A: Reserved word cannot define symbol
Error detected.
Tool error message (example of C compiler)

■ Shortcut menu

There are seven menus below:

- Copy	Copies the selected character string to the clipboard.
- Clear	Clears all the items displayed in the output window.
- Docking View	Sets or resets the docking view.
- Nondisplay	Places the output window in the nondisplay status.
- Set Font	Selects a display font in the output window.
- Keyword Coloring	Whether or not to highlight the error number of the error message is set.
- Logging	The result of output to the Output window is stored in a file.

■ Operation

- Copy

1. Select the character string you want to copy.
2. Click the right button of the mouse in the output window to display the shortcut menu.
3. Select [Copy] from the shortcut menu.

   See Section "4.3.2 Cut, Copy, Paste, Delete".

- Clear

1. Click the right button of the mouse in the output window to displays the shortcut menu.

2. Select [Clear] from the shortcut menu.

- Docking View

1. Click the right button of the mouse in the output window to display the shortcut menu.
2. Select [Docking View] from the shortcut menu.
   - The output window is docked with the frame in the check status.

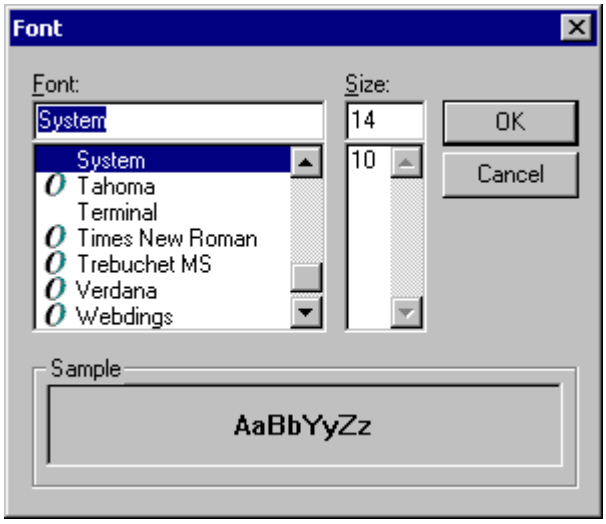
- Display

1. Click the right button of the mouse in the output window to display the shortcut menu.
2. Select [Nondisplay] from the shortcut menu.
   - The output window enters the nondisplay status.

- Set Font

1. Click the right button of the mouse in the output window to display the shortcut menu.
2. Select [Set Font] from the shortcut menu.
   The font selection dialog box (Figure 3.5-2) opens.
3. Select [Font] and [Size], then click the [OK] button.

Figure 3.5-2 Font Setting Dialog Box



- Highlight Keyword

1. Display the shortcut menu by clicking the right button on the Output window.
2. Select [Keyword Coloring].
   - In the check state, the keyword (error number) of the error message is displayed in red.

- Logging

1. Display the shortcut menu by clicking the right button on the Output window.
2. Select [Logging]-[Start] and then specify the file name of the record target file.
   The result of output to the Output window is recorded in the specified file.
3. To suspend logging, select [Logging]-[Record] and uncheck the checkbox.
   Logging is restarted again when the checkbox is checked.
4. To stop logging, select [Logging]-[Termination].

3.6 Edit Window (Standard Editor)

The edit window is used to display and edit a source file and document file, etc.

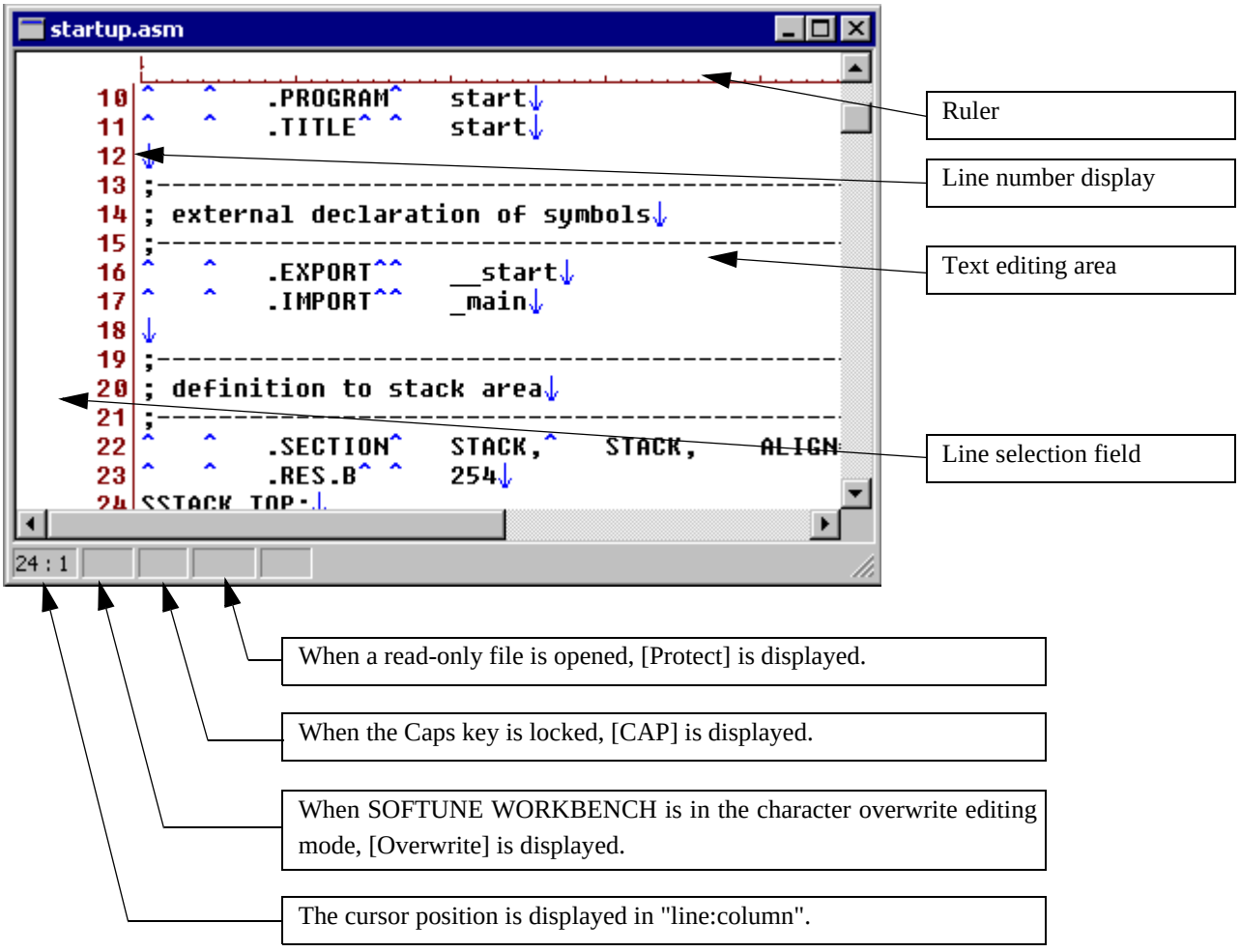
■ Edit window

An example of the edit window docked with the Main Window is given in Figure 3.6-1.

As shown in this example, the edit window docked with the Main Window can be displayed only when the standard editor built into SOFTUNE WORKBENCH is in use.

SOFTUNE WORKBENCH can register and use any external editor. In this case, however, the edit window is not treated as a SOFTUNE WORKBENCH screen.

Figure 3.6-1 Example of Edit Window



■ Shortcut menu

- Source Window — Opens the Source Window.
- Undo — Undoes the last editing.
- Redo — Redoes the canceled editing.
- Cut — Moves the selected character string to the clipboard.

- Copy Copies the selected character string to the clipboard.
- Paste Pastes the character string in the clipboard to the cursor position.
- Delete Deletes the selected character string.
- Select All Selects all characters in the buffer.
- Print Prints the file.
- Add to Project Adds the file being edited to the active project.
- Find Opens the character string find dialog box.
- Replace Opens the character string replacement dialog box.
- Search File See Section "4.3.5 Find in Files".
- Jump to Specified Line Opens the jump line number specification dialog box.
- Bookmark See Section "4.3.7 Bookmark".
- Set Standard Editor See Section "3.6.1 Setting Standard Editor".
- Status Bar Switches status bar display for each window.
- Property Display file information (properties).

- Source Window

1. Click the right button of the mouse to display the shortcut menu, then select [Source Window] from the menu.
   - When SOFTUNE WORKBENCH is in the debug session, the Source Window is displayed. See Section "3.7 Source Window".

     When SOFTUNE WORKBENCH is not in the debug session, this menu cannot be selected

- Undo

1. Click the right button of the mouse to display the shortcut menu, then select [Undo] from the menu.
   - SOFTUNE WORKBENCH cancels the last editing and undoes the status before the editing. See Section "4.3.1 Undo, Redo".

- Redo

1. Click the right button of the mouse to display the shortcut menu, then select [Redo] from the menu.
   - SOFTUNE WORKBENCH cancels the cancel of the last editing.

- Cut

1. Select the character string you want to cut.
2. Click the right button of the mouse to display the shortcut menu, then select [Move] from the menu.
   - SOFTUNE WORKBENCH deletes the selected character string from the edit window and moves it to the clipboard. See Section "4.3.2 Cut, Copy, Paste, Delete".

- Copy

1. Select the character string you want to copy.
2. Click the right button of the mouse to display the shortcut menu, then select [Copy] from the menu.
   - SOFTUNE WORKBENCH copies the selected character string to the clipboard. See Section "4.3.2 Cut, Copy, Paste, Delete".

- Paste

1. Move the cursor to the position into which the character string is to be inserted.
2. Click the right button of the mouse to display the shortcut menu, then select [Paste] from the menu.
   - SOFTUNE WORKBENCH inserts the character string in the clipboard into the cursor position. See Section "4.3.2 Cut, Copy, Paste, Delete".

- Delete

1. Select the character string you want to delete.
2. Click the right button of the mouse to display the shortcut menu, then select [Delete] from the menu.
   - SOFTUNE WORKBENCH deletes the selected character string. See Section "4.3.2 Cut, Copy, Paste, Delete".

- Select All

1. Click the right button of the mouse to display the shortcut menu, then select [All select] from the menu.
   - SOFTUNE WORKBENCH selects all the texts in the edit window and displays them in reverse video. See Section "4.3.3 Select All".

- Print

1. Click the right button of the mouse to display the shortcut menu, then select [Print] from the menu.
   - The print dialog box opens. See Section "4.2.9 Print".

- Add to Project

1. Click the right button of the mouse to display the shortcut menu, then select [Add to Project] from the menu.
   - SOFTUNE WORKBENCH adds the file being edited to the project.

- Find/Replace

1. Click the right button of the mouse to display the shortcut menu, then select [Find] or [Replace] from the menu.
   - SOFTUNE WORKBENCH sets the clipboard character string in [Find What] and opens the Find or Replace dialog box. See Section "4.3.4 Find/Replace".

- Find in Files

1. Click the right button of the mouse to display the shortcut menu, then select [Find in Files] from the menu.
   - SOFTUNE WORKBENCH sets the clipboard character string in [Find what] and opens the Find in Files dialog box. See Section "4.3.5 Find in Files".

- Jump to Specified Line

1. Click the right button of the mouse to display the shortcut menu, then select [Jump to Specified Line] from the menu.

   -The dialog box for specifying the jump destination line opens. See Section "4.3.6 Jump".

   Set the jump destination line number.

   Click the [OK] button.

- Status Bar

  Status Bar switches status bar display and non-display for each Edit Window.

1. Click the right button of the mouse to display the shortcut menu, then select [Status Bar] from the menu.
   - When the status bar has been already displayed, SOFTUNE WORKBENCH switches status bar display to nondisplay and deletes the check mark from the menu.
   - When the status bar is not displayed, SOFTUNE WORKBENCH displays the status bar and adds a check mark to the left end of the menu.

- Property

1. Click the right button of the mouse to display the shortcut menu, then select [Property] from the list.
   - SOFTUNE WORKBENCH displays file information. See Section "4.3.9 Property".

3.6.1 Setting Standard Editor

This section explains how to customize the standard editor.

■ Setting the standard editor

Some standard editor functions can be customized.

● Items that can be customized

- View
  - Line feed mark
  - Tag code mark
  - EOF code mark
  - Line number
  - Ruler
  - Automatic indent
  - Colored display of C keyword
  - Colored display of ASM keyword
  - Colored display of user keyword
  - Colored display of comment statement
- Color
  - Bookmark
  - Error line
  - C keyword
  - Assembly keyword
  - User keyword
  - Comment statement
  - Line Feed
  - Tab
- Font
- Tab count
- Keyword to be highlighted

■ Customization procedure (View)

(1) Click the right button of the mouse in the edit window to display the shortcut menu, then select [Customize...] from the menu.
The standard editor dialog box shown in Figure 3.6-2 opens.

(2) Select the view item you want to set or reset, then click to the left of the item.

(3) When not setting other items, click the [OK] button.
When selecting a color, and so on, set all the related items, then click the [OK] button.

■ Customization procedure (color)

(1) Click the right button of the mouse in the edit window to display the shortcut menu, then select [Customize...] from the menu.
The standard editor dialog box shown in Figure 3.6-2 opens.

(2) Click the color button of [Bookmark], [Error], [C Language], or [ASM Language].
The color selection dialog box opens.

(3) Select a color from the dialog box, then click the [OK] button.
The color of the button changes to the selected color.

(4) When not setting other items, click the [OK] button.
When selecting a font, and so on, set all the related items, then click the [OK] button.

■ Customization procedure (font)

(1) Click the right button of the mouse in the edit window to display the shortcut menu, then select [Customize...] from the menu.
The standard editor dialog box shown in Figure 3.6-2 opens.

(2) Click the [Font] button.
The font selection dialog box opens.

(3) Select a font and size, then click the [OK] button.

(4) When not setting other items, click the [OK] button.
When specifying a tab count, and so on, set all the related items, then click the [OK] button.

■ Customization procedure (tab count)

(1) Click the right button of the mouse in the edit window to display the shortcut menu, then select [Customize...] from the menu.
The standard editor dialog box shown in Figure 3.6-2 opens.

(2) Modify the value displayed in the [Tab Count] field.

(3) When not setting other items, click the [OK] button.
When selecting view items, and so on, set all the related items, then click the [OK] button.

■ Customization procedure (keyword addition)

(1) Click the right button of the mouse in the edit window to display the shortcut menu, then select [Customize...] from the menu.
The standard editor dialog box shown in Figure 3.6-2 opens.

(2) Click [View2] tab to choose the keyword type to be changed from the combo box.
Click [Detail...] button.
The keyword dialog box shown in Figure 3.6-4 opens.

(3) Enter the keyword to be highlighted in [New Item] field. Click [Add] button.

(4) Click the [OK] button.
The keyword files is changed.

(5) When not setting other items, click the [OK] button.
When selecting view items, and so on, set all the related items, then click the [OK] button.

■ Customization procedure (deletion of keyword)

(1) Click the right button of the mouse in the edit window to display the shortcut menu, then select [Customize...] from the menu.
The keyword dialog box shown in Figure 3.6-4 opens.

(2) Click [View2] tab to choose the keyword type to be changed from the combo box. Click [Detail...] button.
The keyword dialog box shown in Figure 3.6-4 opens.

(3) Choose the keyword to be deleted from the [Keyword List]. Click [Delete] button.

(4) Click the [OK] button.
The keyword files is changed.

(5) When not setting other items, click the [OK] button.
When selecting view items, and so on, set all the related items, then click the [OK] button.

■ Customization procedure (initialization of keyword)

(1) Click the right button of the mouse in the edit window to display the shortcut menu, then select [Customize...] from the menu.
The standard editor dialog box shown in Figure 3.6-2 opens.

(2) Click [View2] tab to choose the keyword type to be changed from the combo box. Click [Detail...] button.
The keyword dialog box shown in Figure 3.6-4 opens.

(3) Click [Reset] button.

(4) Click the [OK] button.
The keyword files is changed.

(5) When not setting other items, click the [OK] button.
When selecting view items, and so on, set all the related items, then click the [OK] button.

Figure 3.6-2 Standard Editor Dialog Box

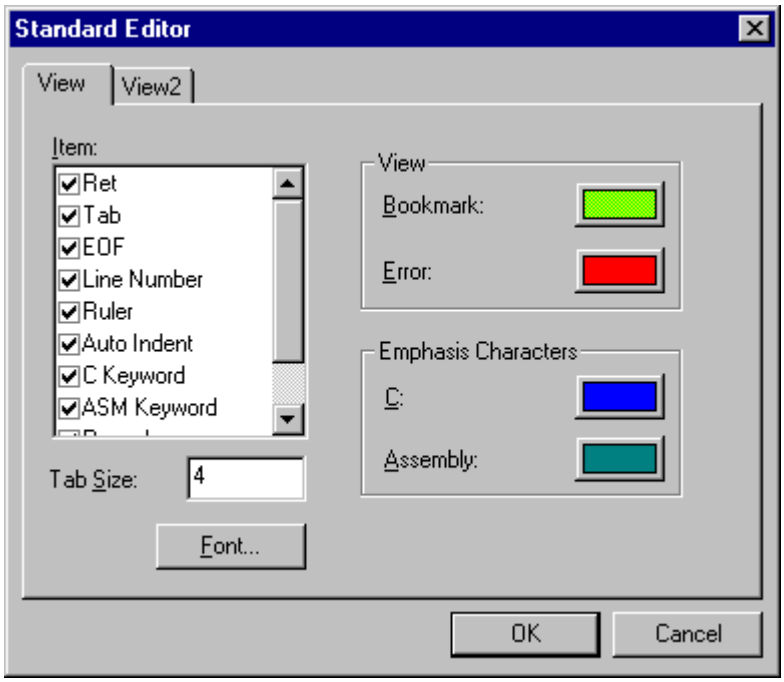


Figure 3.6-3 Standard Editor Dialog Box (view2)

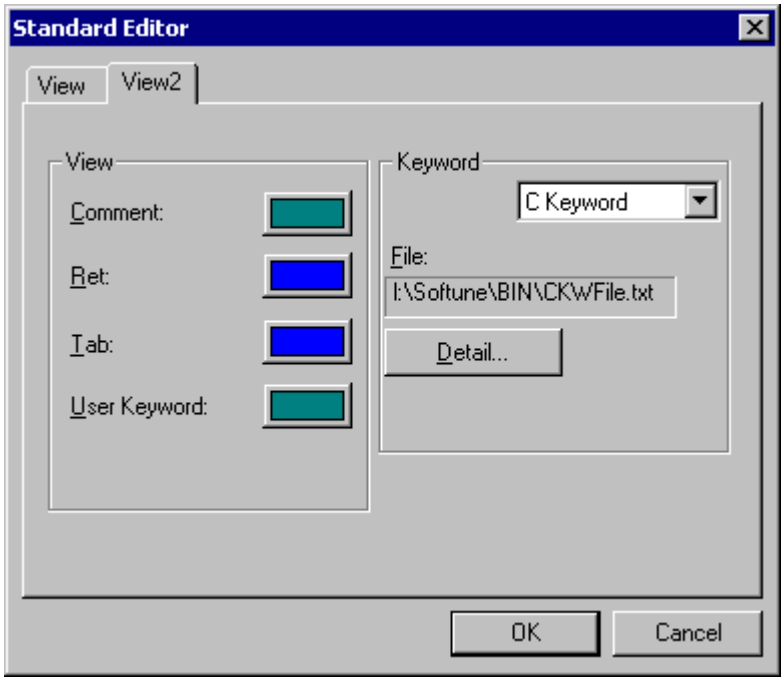
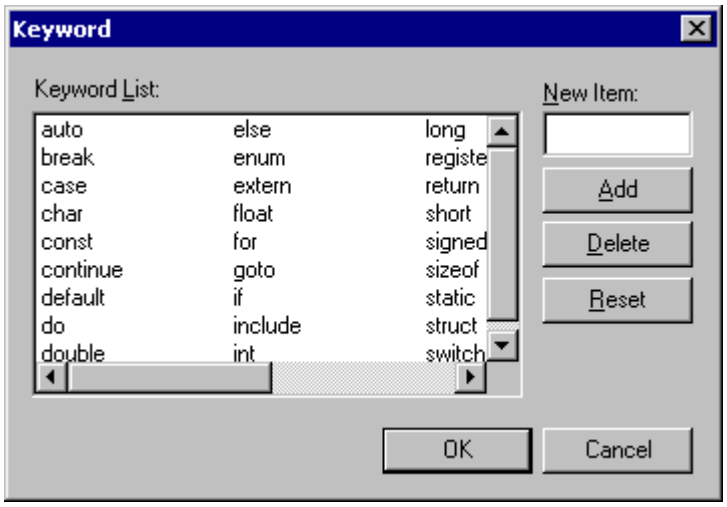


Figure 3.6-4 Keyword Dialog Box



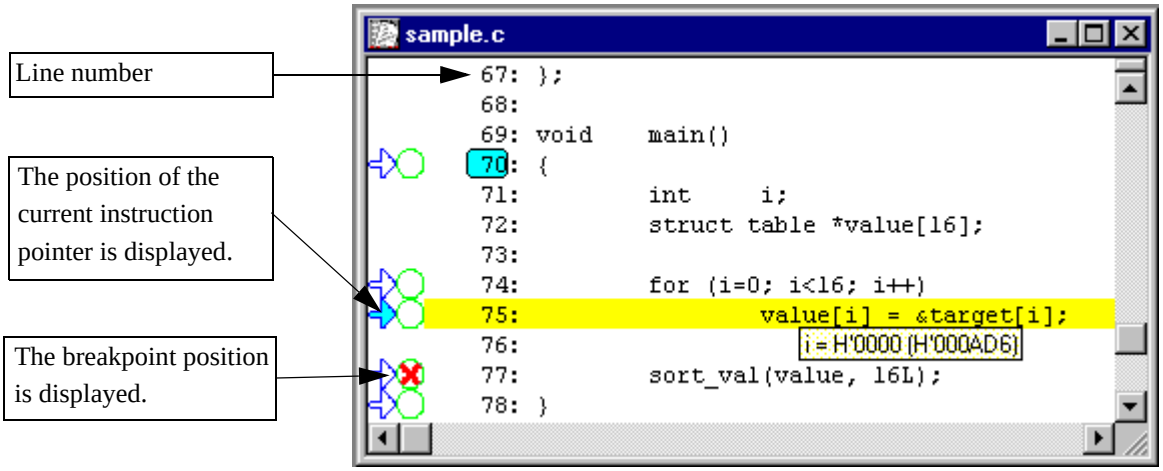
3.7 Source Window

The source window displays the source program at debugging.

■ Source Window

Figure 3.7-1 is an example of the source window.

Figure 3.7-1 Example of Source Window



■ Function

- Breakpoint setting/releasing

  A breakpoint can be set or released by left-clicking '=>❍' of '❍'at the left edge of the source window.

- Instruction execution to specified line (address)

  An instruction can be executed at the specified line (address) by left-clicking of '=>❍' of '=>' at the left edge of the source window. Step execution (step/into) is also enabled by clicking at the PC location.

- Pop-up display of variable value

  By placing cursor of mouse on the name of variable, that variable value can be viewed as pop-up.

- Bookmark setup function

  By setting a bookmark, it enables to view the specified line using search bar or key operation. Please refer to "4.3.7 Bookmark" for details.

- Drag and Drop

  It is possible to drag a character string selected at source window and drop it to memory, disassemble, and watch windows.

  Please refer to description about "drag and drop" at each window for operation done after dropping character string.

- Display coverage

  Coverage can be displayed on a source line basis from the shortcut menu [Display coverage]. In Mix Display, coverage is displayed on a machine instruction basis. Update of the source window is required to display the latest coverage. It is only valid while MB2141/MB2147-01 emulator debugging.

■ Shortcut Menu

There are twelve menus below:

- Edit — Opens the edit window.
- Update — Updates the contents of the specified window. When Display coverage is turned on, the display is updated with the latest coverage. It is only valid while MB2141/MB2147-01 emulator debugging.
- Inline Assemble — See Section "4.4.3 Assembly".
- Find — Opens the character string search dialog box Figure 3.7-2.
- Jump — Displays jump dialog box.
- Go to Current — Displays current PC location.
- Breakpoint Set/Reset — This function is to set or to cancel the breakpoint pointed by cursor.
- Breakpoint Enable/Disable — To make breakpoint in the line pointed by cursor enable or disable.
- Break — See Section "4.6.4 Break Point".
- Watch — Additionally sets watch point
- Stack — See Section "4.6.7 Stack".
- Set coverage — Please refer to Section "4.4.12 Coverage". It is only valid while MB2141/ MB2147-01 emulator debugging.
- Property — Displays source window properties.
- Display coverage — Switches on/off the display of the coverage. When Display coverage is turned on, the executed lines are displayed in green and the unexecuted lines are in brown. It is only valid while MB2141/MB2147-01 emulator debugging.
- Mix Display — Switches whether to display or not disassemble list.
- Activate when stop — Specifies whether to activate the window or not when the execution is stopped.
- Close — Closes source window.
- Edit

  Opens the edit window to modify the current source file.

  See Section "3.6 Edit Window (Standard Editor)".
- Find

  Searches the text in the current source file. Perform this operation in the following procedure:

  1. Click the right button of the mouse in the source window to display the shortcut menu, then select [Find] from the menu.
     The search dialog box shown in Figure 3.7-2 opens.
  2. Set the character string you want to search.
  3. Set a search direction.
  4. If necessary, display a check mark to [Match case].
  5. Click the [Find Next] button.
     The fond character string is displayed in reverse video.
  6. To terminate search, click the [Cancel] button.

Figure 3.7-2 Search Dialog Box in Source Window

- Jump

  Specifies the position to display in the source window. Perform this operation in the following procedure:

  1. Click the right button of the mouse in the source window to display the shortcut menu, then select [Jump] from the menu.
     See Section "4.3.6 Jump".
  2. Specify a specification format.
  3. Select a display position.
  4. Click the [OK] button.

- Breakpoint Set/Reset

  After moving cursor to the line where you want to set or to cancel a breakpoint, click the right button to see short cut menu. Then, please select [Breakpoint Set/Reset].

- Breakpoint Enable/Disable

  After moving cursor to the line where breakpoint is set to be enable or disable, click right button to see the short cut menu. Then, please select [Breakpoint Enable/Disable]. This enables to switch the breakpoint, which is pointed by cursor, enable or disable.

- Watch

  Specifies the variable to watch and opens the watch window. Perform this operation in the following procedure:

  1. Click the right button of the mouse in the source window to display the shortcut menu, then select [Watch] from the menu.
     See Section "4.4.7 Watch".
  2. Specify a variable name.
  3. Select a [mode] as required.
  4. Click the [OK] button.

- Property

  Click the right button of the mouse to display the shortcut menu, then select [Property] from the menu.

  - File information is displayed (see Section "4.3.9 Property").

- Mixed Display

  1. Click the right button of the mouse in the source window to display the shortcut menu.
  2. Check [Mix Display].
     - When a check mark is displayed to the left of [Mix Display], mixed display is already set.
     - When a check mark is not displayed to the left of [Mix Display], mixed display is not set.

3. Select [Mix Display].
   - When it is already set, mixed display is reset and the check mark is deleted.
   - When it is not set, mixed display is set and a check mark is displayed.

- Display coverage

1. Click the right button of the mouse in the source window to display the shortcut menu.
2. Check [Display coverage].
   A check mark attached to the left side of [Display coverage] indicates that Display coverage has already been turned on.
   If a check mark isn't attached to the left side of [Display coverage], Display coverage is turned off.
3. Select [Display coverage].
   - Display coverage is turned off, and the check mark is disappeared when it has already been turned on.
   - Display coverage is turned on, and a check mark is attached when it has been turned off.

Note:

The coverage display in the source window isn't automatically updated. You need to update the display with the latest information using the shortcut menu [Update] or by selecting [Update window] or [Update all windows] from the main menu.

- Activate when stop

  Specify whether the source window should set to active or not when the program is stopped. However, if a source window is displayed as new, it should always be activated. Please operate with following procedure.

  1. Click a right button of source window to display short cut menu.
  2. Please make sure [Activate when stop].
     If a check mark is placed on the left side of [Activate when stop], source file, which corresponds to break address while halting program execution, should be activated. On the other hand, if the source file is not displayed, open a source file that corresponds to break address regardless of specification.

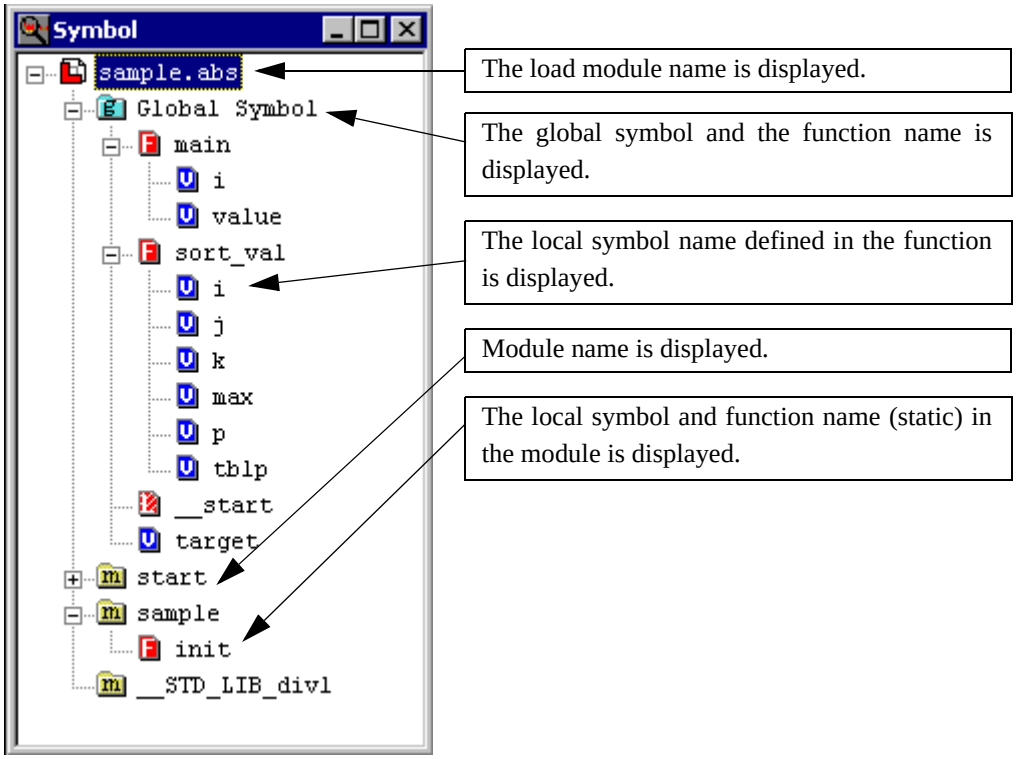
3.8 Symbol Window

The Symbol Window displays the target file name of the current project and the symbol names (e.g., variable name and function name) being used by the file in the tree view format.

■ Symbol Window

Figure 3.8-1 is an example of the Symbol Window.

Figure 3.8-1 Example of Symbol Window



■ Shortcut Menu

Jump	Opens the Source Window of the specified function.
Watch	Sets specified symbol as watch point.
Break	See Section "4.6.4 Break Point".
Properties	Displays symbol properties.
Mangled name	Specify whether to display a symbol name as a mangled name.
Refresh window	The latest symbol information is displayed.
Close	Closes Symbol Window.

- Jump

Click the right button of the mouse on the function name to display the shortcut menu, then select [Jump] from the menu.

The source window in which the function is defined opens.

- Watch

Click the right button of the mouse on the variable or function name to display the shortcut menu, then select [Watch] from the menu. The variable or function is added to the watch and the Watch Window is displayed.

- Property

Click the right button of the mouse to display the shortcut menu, then select [Property] from the menu.

File information is displayed. See Section "4.3.9 Property".

- Mangled name

Click the mouse right button to open the shortcut menu and select [Mangled name]. Switch between display and non-display of a symbol name as a mangled name.

- Refresh window

Click the mouse right button to open the shortcut menu and select [Refresh window]. The latest symbol information is displayed.

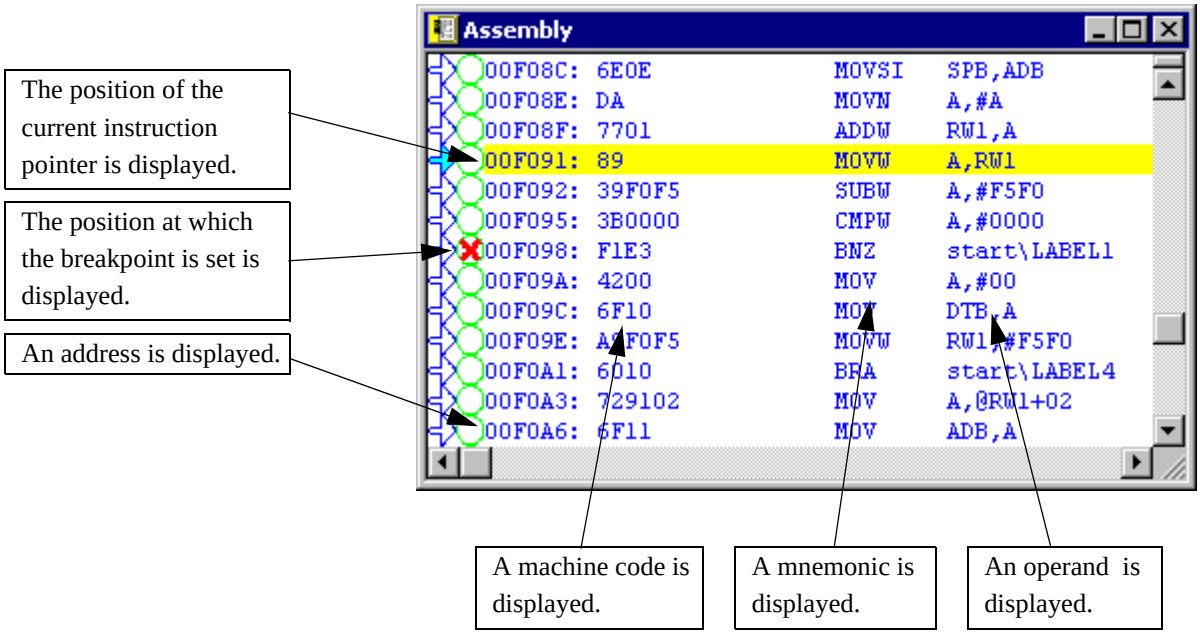
3.9 Disassemble Window

This window is displayed only when SOFTUNE WORKBENCH is in the debug session. The disassemble window displays the result obtained by executing disassembling at the specified address.

■ Disassemble Window

Example of the disassemble window displays at Figure 3.9-1.

Figure 3.9-1 Disassemble Window


Assembly
00F08C: 6E0E MOVSI SPB,ADB
00F08E: DA MOVN A,#A
00F08F: 7701 ADDW RW1,A
00F091: 89 MOVW A,RW1
00F092: 39F0F5 SUBW A,#F5F0
00F095: 3B0000 CMPW A,#0000
00F098: F1E3 BNZ start\LABEL1
00F09A: 4200 MOV A,#00
00F09C: 6F10 MOV DTB,A
00F09E: A8F0F5 MOVW RW1,#F5F0
00F0A1: 6010 BRA start\LABEL4
00F0A3: 729102 MOV A,@RW1+02
00F0A6: 6F11 MOV ADB,A
The position of the current instruction pointer is displayed.
The position at which the breakpoint is set is displayed.
An address is displayed.
A machine code is displayed.
A mnemonic is displayed.
An operand is displayed.

■ Function

- Breakpoint setting/releasing

  A breakpoint can be set or released by left-clicking '=>❍' of '❍' at the left edge of the disassemble window.

- Instruction execution at specified line (address)

  An instruction can be executed at the specified line (address) by left-clicking of '=>❍' of '❍'at the left edge of the disassemble window. Step execution (step/into) is also enabled by clicking the left mouse button at the PC location.

- Drag and Drop

  By dropping function name, label, or address from source window to disassemble window, display will be jumped to a location where an address of dropped character string is shown.

- Display coverage

  Coverage can be displayed on a machine instruction basis from the shortcut menu [Display coverage]. It is only valid while MB2141/MB2147-01 emulator debugging.

■ Shortcut Menu

There are ten menus below:

Inline Assemble	See Section "4.4.3 Assembly".
Jump	Displays jump dialog box.
Go to Current	Displays current PC location.
Breakpoint Set/Reset	This function is to set or to cancel the breakpoint pointed by cursor.
Breakpoint Enable/Disable	To make breakpoint in the instruction pointed by cursor enable or disable.
Break	See Section "4.6.4 Break Point".
Watch	See Section "4.4.7 Watch".
Stack	See Section "4.6.7 Stack".
Set coverage	Please refer to Section "4.4.12 Coverage". It is only valid while MB2141/ MB2147-01 emulator debugging.
Display coverage	Switches on/off the display of the coverage. When Display coverage is turned on, the executed lines are displayed in green and the unexecuted lines are in brown. It is only valid while MB2141/MB2147-01 emulator debugging.
Activate when stop	Specifies whether to activate the window or not when the execution is stopped.
Close	Closes disassemble window.

- Jump

  Specifies the position to display in the disassemble window. Perform this operation in the following procedure:

  1. Click the right button of the mouse in the disassemble window to display the shortcut menu, then select [Jump] from the menu.
     The jump dialog box opens. (See Section "4.3.6 Jump").
  2. Select a specification format.
  3. Specify a cursor display position.
  4. Click the [OK] button.

- Display coverage

  1. Click the right button of the mouse in the disassemble window to display the shortcut menu.
  2. Check [Display coverage].
     A check mark attached to the left side of [Display coverage] indicates that Display coverage has already been turned on.
     If a check mark isn't attached to the left side of [Display coverage], Display coverage is turned off.
  3. Select [Display coverage].
     - Display coverage is turned off, and the check mark is disappeared when it has already been turned on.
     - Display coverage is turned on, and a check mark is attached when it has been turned off.

- Breakpoint Set/Reset

  After moving cursor to the instruction where you want to set or to cancel a breakpoint, click the right button to see short cut menu. Then, please select [Breakpoint Set/Reset].

- Breakpoint Enable/Disable

  After moving cursor to the instruction where breakpoint is set to be enable or disable, click right button to see the short cut menu. Then, please select [Breakpoint Enable/Disable]. This enables to switch the breakpoint, which is pointed by cursor, enable or disable.

- Activate when stop

  Specify whether the disassemble window should set to active or not when the program is stopped. However, if a disassemble window is displayed as new, it should always be activated. Please operate with following procedure.

  1. Click a right button of disassemble window to display short cut menu.
  2. Please make sure [Activate when stop].

3.10 Register Window

The Register Window is used to display and modify MCU register values during debugging. This window is displayed only when SOFTUNE WORKBENCH is in the debug session.

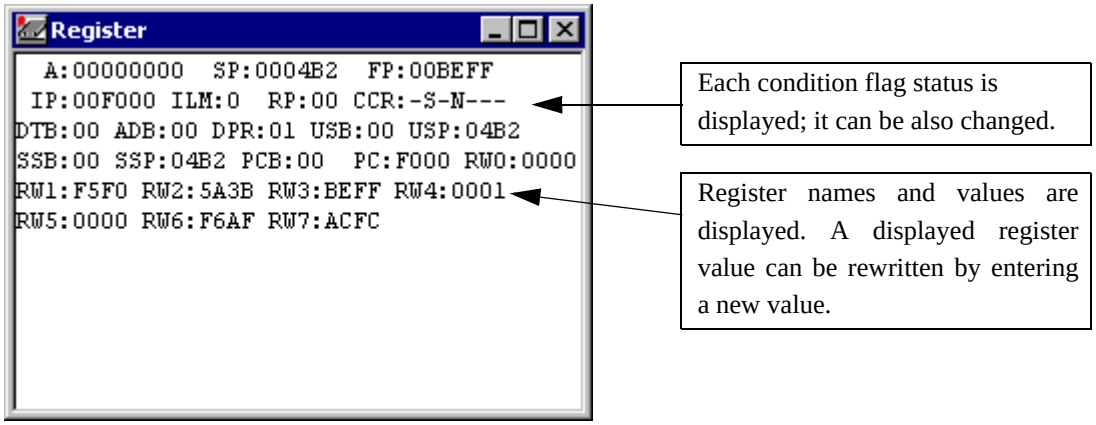
■ Register Window

Figure 3.10-1 is an example of the Register Window. When debugging, the Register Window displayed the register values. Register values can be rewritten directly. To rewrite values using expressions and symbols, etc., select [Edit] from the shortcut menu for the Register Window.

Arrange registers according to the size of the Register Window.

To select the registers to be displayed, select [Display Setup] from the shortcut menu for the Register Window.

Figure 3.10-1 Example of Register Window


Register
A:00000000 SP:0004B2 FP:00BEFF
IP:00F000 ILM:0 RP:00 CCR:-S-N---
DTB:00 ADB:00 DPR:01 USB:00 USP:04B2
SSB:00 SSP:04B2 PCB:00 PC:F000 RW0:0000
RW1:F5F0 RW2:5A3B RW3:BEFF RW4:0001
RW5:0000 RW6:F6AF RW7:ACFC
Each condition flag status is displayed; it can be also changed.
Register names and values are displayed. A displayed register value can be rewritten by entering a new value.

■ Shortcut Menu

- Display Setup See Section "3.10.1 Setting Register Display".
- Edit Displays Register Edit Dialog Box.
- Close Closes Register Window.

- Edit

  Edits register values. Beside this operation, the numerical values displayed in the Register Window can be directly edited. See Section "4.4.4 Register".

Click the right button of the mouse in the Register Window to display the shortcut menu, then select [Edit] from the menu.
The register edit dialog box opens.

Select a register name.

Enter the value you want to set.

Click the [OK] button.

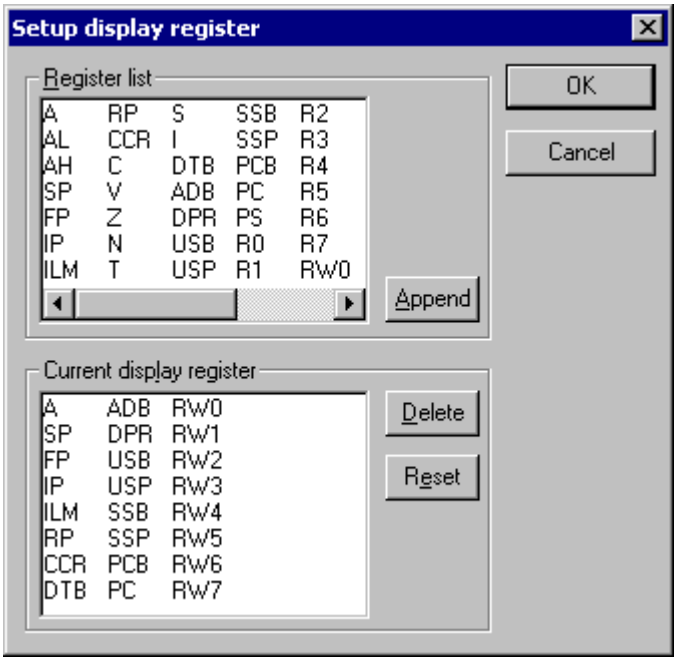
3.10.1 Setting Register Display

This section explains how to set register display.

■ Setting register display

Set the registers to be displayed in the Register Window in the following procedure:

Figure 3.10-2 Display Register Setting Dialog Box



■ Adding registers

Add the register you want to display in the following procedure:

Click the right button of the mouse in the Register Window to display the shortcut menu, then select [Display Setup] from the menu.
The display setting dialog box shown in Figure 3.10-2 opens.

Select the register you want to display from [Register List] (upper part of display register setting dialog box), then click the [Add] button.
The selected register is set in [Current Display Register] (lower part).

Set all the registers you want to display by repeating the above operation.

Click the [OK] button.

■ Deleting registers

Select the register you want to delete in the following procedure:

Click the right button of the mouse in the Register Window to display the shortcut menu, then select [Display Setup] from the menu.
The display setting dialog box shown in Figure 3.10-2 opens.

Select the register you want to delete from [Register List] (lower part of display register setting dialog box), then click the [Delete] button.
The selected register is deleted from [Current Display Register] (lower part).

Set all the registers you want to delete by repeating the above operation.

Click the [OK] button.

■ Restoring registers to the initial status

Restore the registers in [Current Display Registers] to the initial status (status when SOFTUNE WORKBENCH was installed) in the following procedure:

Click the right button of the mouse in the Register Window to display the shortcut menu, then select [Display Setup] from the menu.
The display setting dialog box shown in Figure 3.10-2 opens.

Click the [Reset] button.
The registers in [Current Display Registers] (lower part of display register setting dialog box) are restored to the initial status.

Click the [OK] button.

3.11 Memory Window

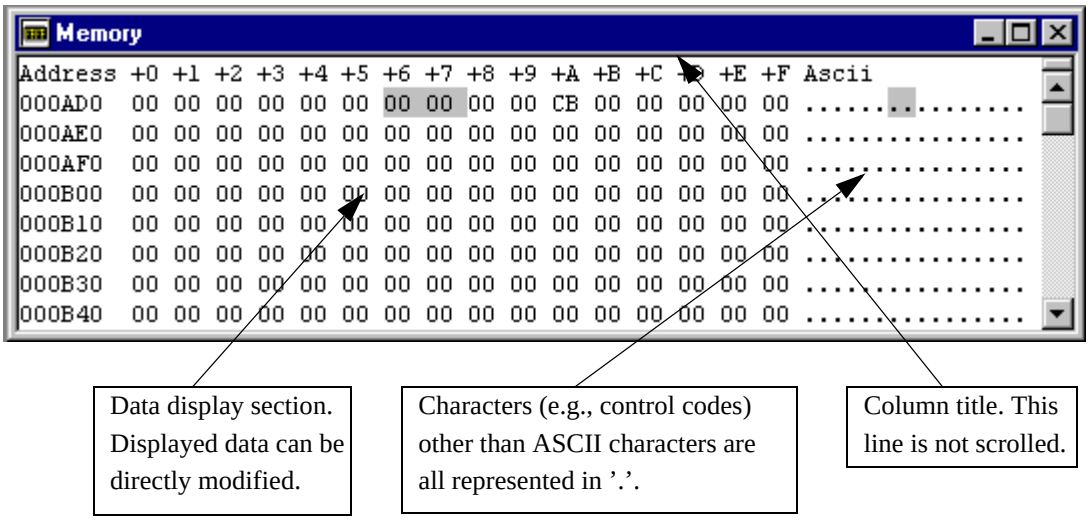
This window opens only when SOFTUNE Workbench is in the debug session. The memory window is used to display and modify memory data at the specified address.

■ Memory Window

Figure 3.11-1 is an example of the memory window. When debugging memory values can be displayed. The register values can be directly rewritten.

To modify by specifying the expression and symbol in the memory values, select [Edit] from the shortcut menu for the memory window.

Figure 3.11-1 Example of Memory Window



Data display section. Displayed data can be directly modified.

Characters (e.g., control codes) other than ASCII characters are all represented in '.'.

Column title. This line is not scrolled.

■ Function

- Display the change

  The location, where was changed from the previous contents, will be shown in red while program execution is stopped or monitoring. This function is only valid in a range where currently displayed by memory window.

- Display with 16 bytes fixed in size

  The number of bytes at one line can be chosen: [Auto] which is corresponding to the size of window or [4byte], [8byte], [16byte], [32byte] and [64byte] which the size of bytes are fixed. In order to set up the number of bytes, use [Setup] at short-cut menu.

- Bookmark setup function

  Bookmark can be set by coloring the specified address range. By setting a bookmark, it enables to view the specified line using search bar or key operation. Please refer to "4.3.7 Bookmark" for details.

- Drag and Drop

  By dropping variable name, function name, label or address from source window to memory window, the address of dropped character string is shown and address range will be chosen.

■ Shortcut Menu

There are fourteen menus below:

- Compare	See Section "4.4.5 Memory".
- Find	See Section "4.4.5 Memory".
- Special	See Section "4.4.5 Memory".
- Inline Assemble	See Section "4.4.3 Assembly".
- Jump	See Section "4.3.6 Jump".
- Bookmark	See Section "4.4.5 Memory".
- Edit	See Section "4.4.5 Memory".
- Setup	See Section "4.4.5 Memory".
- Inaccessible area	Displays the [Inaccessible area] tab in the [Debug environment] of the [Setup] menu.
- Break	See Section "4.6.4 Break Point".
- Event	See Section "4.6.5 Event".
- Watch	See Section "4.4.7 Watch".
- Monitoring	Switches whether to enable or disable monitoring.
- Close	Closes the memory window.

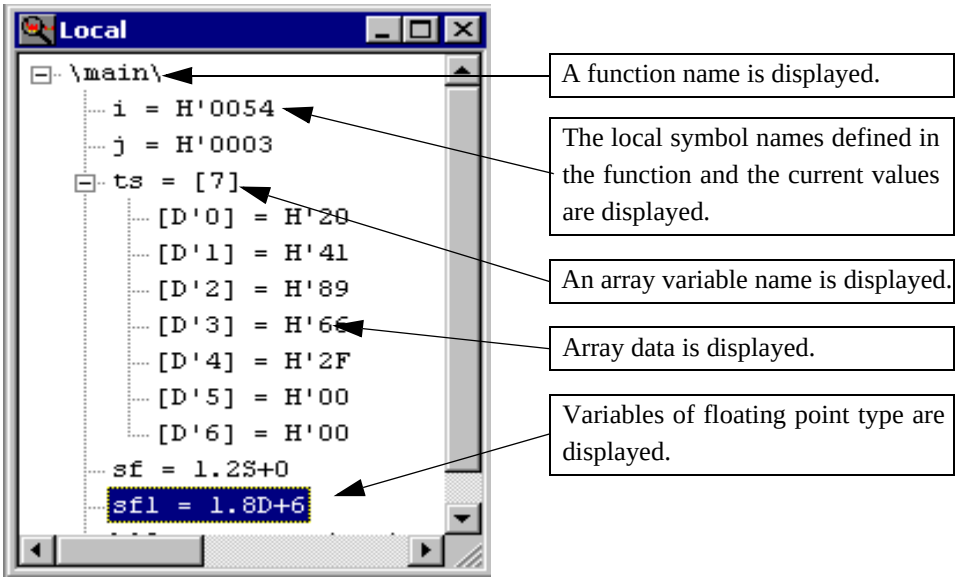
3.12 Local Variable Window

This window is displayed only when SOFTUNE WORKBENCH is in the debug session. The local variable window is used to display and modify local variable values. Local variables are displayed in the tree format (function name root).

■ Local Variable Window

Figure 3.12-1 is an example of the local variable window.

Figure 3.12-1 Example of Local Variable Window



■ Shortcut Menu

There are five menus below:

- Radix — See Section "4.4.6 Local".
- Edit — See Section "4.4.6 Local".
- Memory window — Display the memory window and setup bookmark.
- Property — Displays symbol properties.
- Close — Closes local window.
- Memory window

  When selected, the following two submenus are opened:
- Jump

  To display memory window by using selected address of variable.
- Add bookmark

  By using selected variable address, bookmark will be added to memory window.
- Property

  Information for the variable is displayed. For details, see Section "4.3.9 Property".

3.13 Watch Window

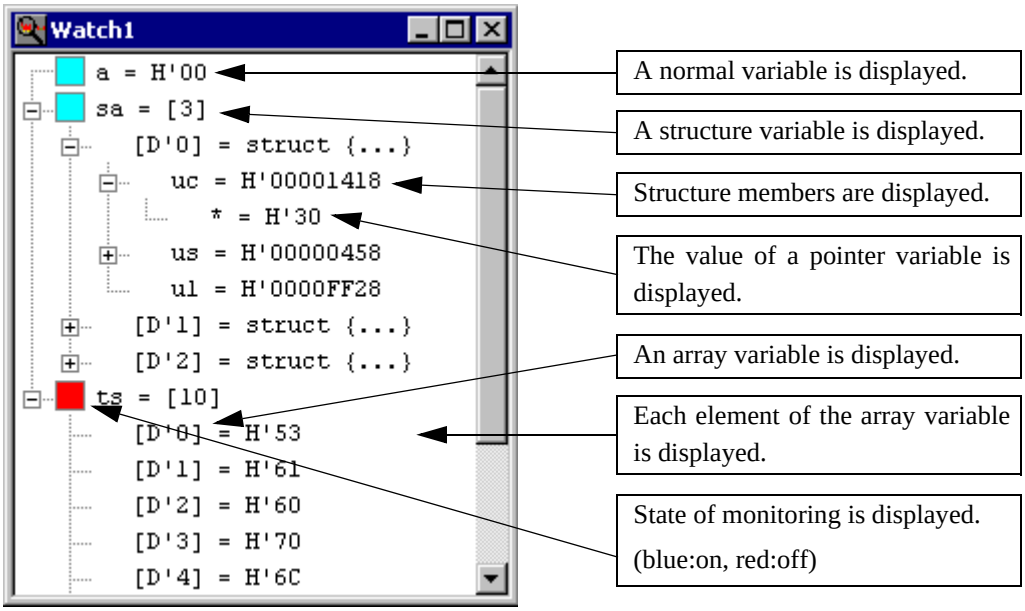
This window is displayed only when SOFTUNE WORKBENCH is in the debug session. The watch window is used to display and modify the values of the specified variables. Variables are displayed in the tree format.

■ Watch Window

The watch window can be opened independent 4-windows. When being registered a variable, user select watch window no (1,2,3 or 4). Because the watch window can be registered the variables for each window.

Figure 3.13-1 is an example of the watch window.

Figure 3.13-1 Example of Watch Window



■ Function

- Drag and Drop

  1. Sort variables in the window

     If you register multiple variables in the window, you can change the order of the variables by dragging a variable and dropping it into a desired place.

  2. Cooperation between the watch windows

     You can copy/move the variables by dragging a registered variable and dropping it into another watch window. If you drag a structure member, only the member is copied to another window, and the structure remains in the original window.

- Multiple selection of variables

  You can select multiple variables at a time for the batch operation. Available operations are as follows:

  - Copy/Move by drag and drop
  - Setting of the radix

- Delete
- On/Off of monitoring

■ Shortcut Menu

There are twelve menus below:

- Radix — See Section "4.4.7 Watch". If multiple variables are selected, they are deleted all together.
- Set — See Section "4.4.7 Watch".
- Element — Specify range of displayed array element.
- Edit — See Section "4.4.7 Watch".
- Delete — See Section "4.4.7 Watch". If multiple variables are selected, they are deleted all together.
- All Delete — Deletes all variables being stored from the window.
- Inaccessible area — Call up the [Inaccessible area] tab in the [Debug environment] of the [Setup] menu.
  - Monitoring — Switches whether to enable or disable monitoring.
  - Individual setting — The each variable which is registered, user sets "ON" or "OFF" of monitoring. User distinguishes between "ON" and "OFF" by a color of square ICON. A blue means on, a red means off. If multiple variables are selected, they are deleted all together.
  - Memory window — Display the memory window and setup bookmark.
  - Property — Displays watch properties.
  - Close — Closes watch window.
  - Memory window

    When selected, the following two submenus are opened:

    - Jump

      To display memory window by using selected address of variable.

    - Add bookmark

      By using selected variable address, bookmark will be added to memory window.

- Property

1. Click the right button of the mouse on a displayed variable name to display the shortcut menu, then select [Property] from the menu.
   - Information for the variable is displayed. For details, see Section "4.3.9 Property".
2. Click the [Close] button

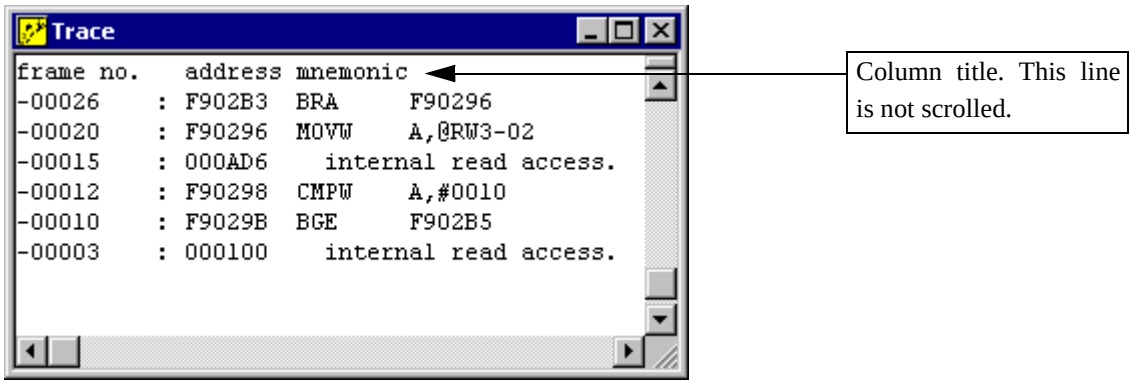
3.14 Trace Window

This section explains the Trace Window. This window is displayed only when SOFTUNE WORKBENCH is in the debug session.

■ Trace Window

The Trace Window displays the trace result. For the trace function, see Section "4.4.8 Trace". Figure 3.14-1 is an example of the Trace Window

Figure 3.14-1 Example of Trace Window


Trace
frame no. address mnemonic
-00026 : F902B3 BRA F90296
-00020 : F90296 MOVW A,@RW3-02
-00015 : 000AD6 internal read access.
-00012 : F90298 CMPW A,#0010
-00010 : F9029B BGE F902B5
-00003 : 000100 internal read access.
Column title. This line is not scrolled.

■ Shortcut Menu

- Event mode	See Section "4.4.8 Trace".
- Refresh	Refresh Trace Window.
- Start	See "4.4.8 Trace".
- Abort	See "4.4.8 Trace".
- Jump	See Section "4.3.6 Jump".
- Back Trace	See Section "4.4.8 Trace".
- Instruction	Displays trace result in instruction mode (Valid when the emulator debugger is used).
- Cycle	Displays trace result in cycle mode (Valid when the emulator or simulator debugger is used).
- Source	Displays trace result in source mode (Valid when the emulator or simulator debugger is used).
- Measurement point	See "4.4.8 Trace".
- Event hit information	See "4.4.8 Trace".
- Setup	See Section "4.4.8 Trace".
- Find	See Section "4.4.8 Trace".
- Save	See Section "4.4.8 Trace".
- Clear	Clears display mode trace buffer.

- Close Closes Trace Window.
- Instruction/cycle/source
  1. Click the right button of the mouse to display the shortcut menu.
  2. Select [Instruction], [Cycle], or [Source] from the shortcut menu.
     The trace result is displayed in the selected mode and a check mark is displayed to the left of the selected submenu.

Note:

[Start], [Abort], [Measurement point], and [Event hit information] in the shortcut menu become valid only when the debug function of MB2147-01 is in Trace Enhancement mode. For more information, please refer to Section "2.3.1.6 Debug Function" in SOFTUNE Workbench User's Manual.

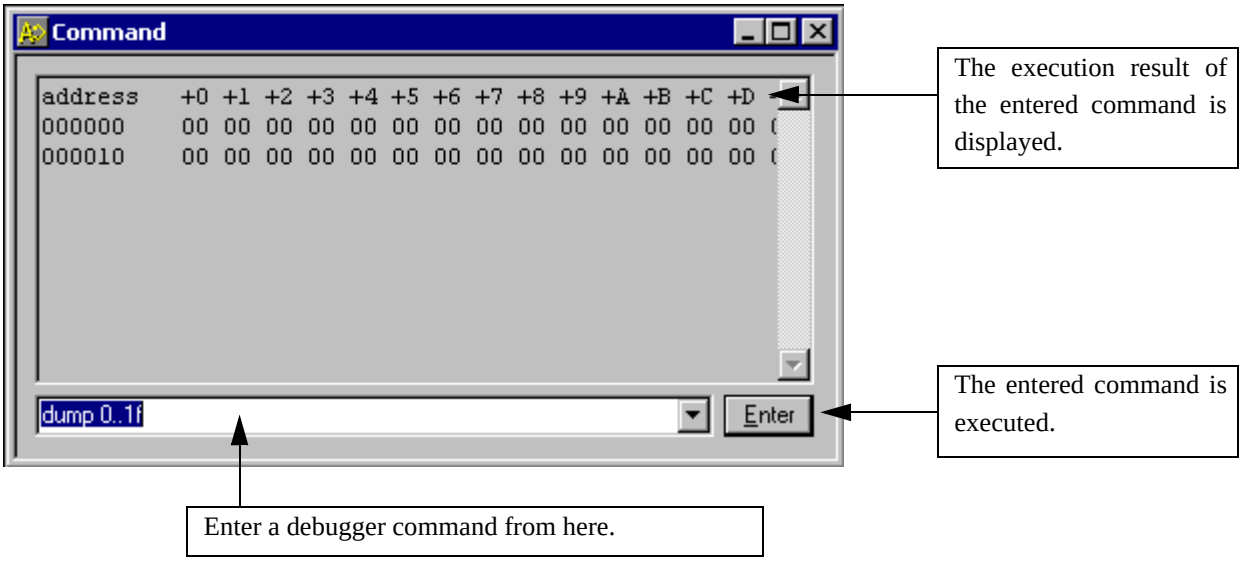
3.15 Command Window

The Command Window displays the execution result of the directly entered debugger command. This window is displayed only when SOFTUNE WORKBENCH is in the debug session.

■ Command Window

Figure 3.15-1 is an example of the Command Window.

Figure 3.15-1 Example of Command Window


Command
address +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D
000000 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000010 00 00 00 00 00 00 00 00 00 00 00 00 00 00
dump 0..1f
Enter
The execution result of the entered command is displayed.
The entered command is executed.
Enter a debugger command from here.

■ Shortcut Menu (Command input field)

- Undo	Cancels the immediately preceding editing.
- Cut	Moves the selected character string to the clipboard.
- Copy	Copies the selected character string to the clipboard.
- Paste	Pastes the character string in the clipboard.
- Delete	Deletes the selected character string.
- All select	Selects all the entered character strings.

- Undo

Click the right button of the mouse on the command input field to display the shortcut menu.

Select [Undo] from the shortcut menu.

- Cut

Select the character string you want to cut.

Click the right button of the mouse on the command input field to display the shortcut menu.

Select [Cut] from the shortcut menu.

- Copy

Select the character string you want to copy.

Click the right button of the mouse on the command input field to display the shortcut menu.

Select [Copy] from the shortcut menu.

- Paste

Click the right button of the mouse on the command input field to display the shortcut menu.

Select [Paste] from the shortcut menu.

- Delete

Select the character string you want to delete.

Click the right button of the mouse on the command input field to display the shortcut menu.

Select [Delete] from the shortcut menu.

- All select

Click the right button of the mouse on the command input field to display the shortcut menu.

Select [All select] from the shortcut menu.

■ Shortcut Menu (Other than the command input field)

- Specify Batch file	Opens the file dialog box for specifying the batch file.
- Specify Alias File	Opens the file dialog box for specifying the alias file.
- Character String Replace Setup	Displays Command Replacement Dialog Box.
- Logging	Sets log start, log status display and log end.
- Close	Closes Command Window.

- Specify Batch File

Click the right button of the mouse to display the shortcut menu.

Select [Specify Batch File] from the shortcut menu.
The [Open File] file dialog box opens.

Select a batch file from the dialog box, then click the [Open] button.

- Specify Alias File

Click the right button of the mouse to display the shortcut menu.

Select [Specify Alias File] from the shortcut menu.
The [Open File] file dialog box opens.

Select an alias file from the dialog box, then click the [Open] button.

- Set Character String Replacement

Click the right button of the mouse to display the shortcut menu.

Select [Set Character String Replacement] from the shortcut menu.
The [Command Replacement] dialog box opens. For details, see Section "3.15.1 Setting Character String Replacement".

- Logging

For logging, see Section "3.15.2 Logging".

- Close

Click the right button of the mouse to display the shortcut menu.

Select [Close] from the shortcut menu.

3.15.1 Setting Character String Replacement

This section explains how to set character string replacement in the Command Window.

■ Setting Character String Replacement

The aliases, variables, and macro names of the commands used in the Command Window can be defined.

■ Alias

The commands to be used often can be started simply by specifying their names. Set and delete aliases in the following procedures:

- Adding an alias

Click the right button of the mouse in the Command Window to display the shortcut menu.

Select [Set Character String Replacement] from the shortcut menu.
The character string replacement setting dialog box shown in Figure 3.15-2 opens.

Set an alias.
Specify the alias that does not conflict with command names and the aliases that has been already registered.

Write the command to be assigned to the specified alias.
The command can be written, including its parameters.

Click the [Append] button.

- Deleting an alias

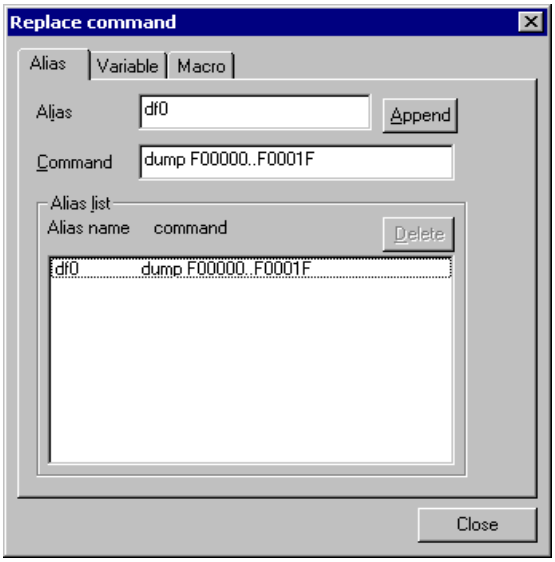
Click the right button of the mouse in the Command Window to display the shortcut menu.

Select [Set Character String Replacement] from the shortcut menu.
The character string replacement setting dialog box shown in Figure 3.15-2 opens.

Select the alias to be deleted from the alias list.

Click the [Delete] button.

Figure 3.15-2 Character String Replacement Setting Dialog Box



■ Variable

The name assigned to the address range to be referenced often can be used for command input. Add and delete a variable in the following procedures

- Adding a variable

Click the right button of the mouse in the Command Window to display the shortcut menu.

Select [Set Character String Replacement] from the shortcut menu.
The character string replacement setting dialog box shown in Figure 3.15-2 opens.

Click the [Variable] tag.
The variable setting dialog box shown in Figure 3.15-3 opens.

Set a variable.
Specify the variable name that does not conflict with the variable names that have been already registered.

Write the character string to be assigned to the specified variable name.

Click the [Append] button.

- Deleting a variable

Click the right button of the mouse in the Command Window to display the shortcut menu.

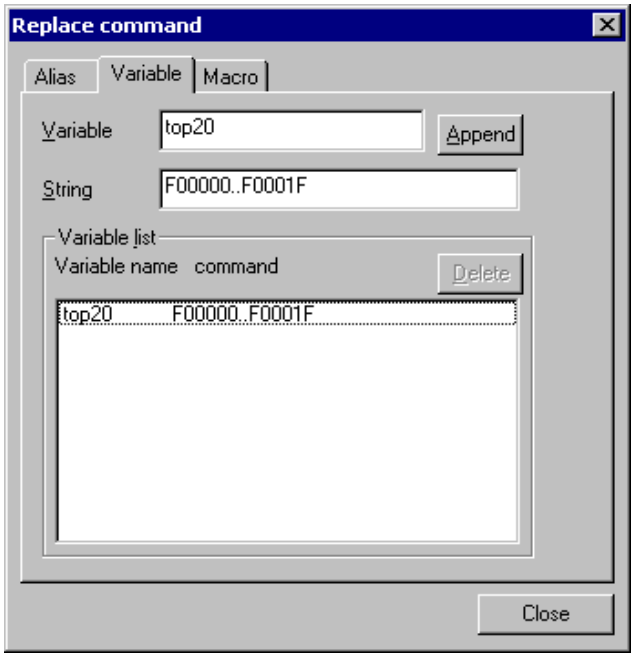
Select [Set Character String Replacement] from the shortcut menu.
The character string replacement setting dialog box shown in Figure 3.15-2 opens.

Click the [Variable] tag.
The variable setting dialog box shown in Figure 3.15-3 opens.

Select the variable name to be deleted from the debug variable list.

Click the [Delete] button.

Figure 3.15-3 Variable Setting Dialog Box



■ Macro

A combination of the special commands to be used often can be named and used as a macro. Add and delete a macro in the following procedures:

- Adding a macro

Click the right button of the mouse in the Command Window to display the shortcut menu.

Select [Set Character String Replacement] from the shortcut menu.
The character string replacement setting dialog box shown in Figure 3.15-2 opens.

Click the [Macro] tag.
The macro setting dialog box shown in Figure 3.15-4 opens.

Set a macro name.
Specify a macro name that does not conflict with the macro names that have been already registered.

Write the commands to be assigned to the specified macro name.
Aliases, variables, and other macros can be also written.

Click the [Append] button.

- Deleting a macro

Click the right button of the mouse in the Command Window to display the shortcut menu.

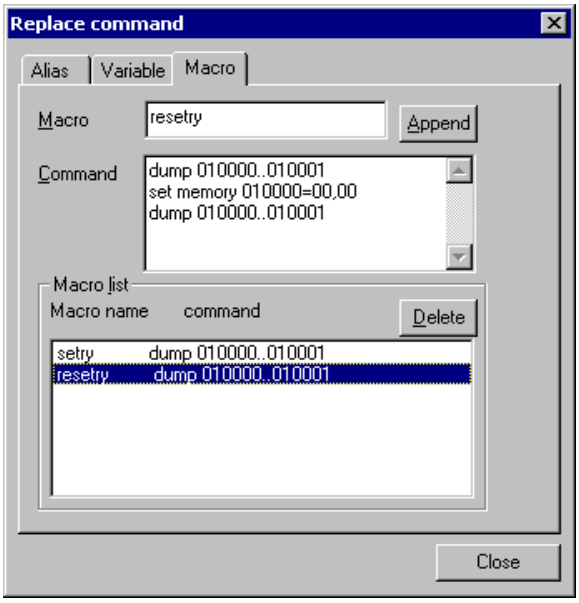
Select [Set Character String Replacement] from the shortcut menu.
The character string replacement setting dialog box shown in Figure 3.15-2 opens.

Click the [Macro] tag.
The macro setting dialog box shown in Figure 3.15-4 opens.

Select the macro name to be deleted from the macro list.

Click the [Delete] button.

Figure 3.15-4 Macro Setting Dialog Box



3.15.2 Logging

This section explains SOFTUNE WORKBENCH logging.

■ Logging

To record a command execution log in the Command Window in a file is called logging. To control logging, click the right button of the mouse in the Command Window to display the shortcut menu, then select [Logging].

Selecting [Logging] from the shortcut menu displays the following submenus:

- Start
- State
- Exit

- Start

Click the right button of the mouse to display the shortcut menu.

Select [Start] from the shortcut menu.
The file dialog box for selecting a logging file opens.

Specify a logging file name, then click the [Save] button.

- State

Click the right button of the mouse to display the shortcut menu.

Select [State] from the shortcut menu.
The logging state display dialog box shown in Figure 3.15-5 opens.

If necessary, change the state and output message control.

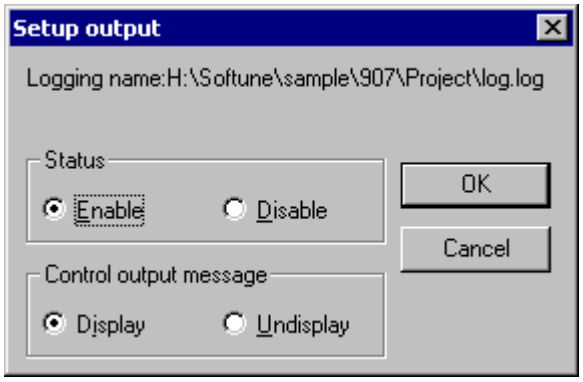
Click the [OK] button.

- Exit

Click the right button of the mouse to display the shortcut menu.

Select [Exit] from the shortcut menu.

Figure 3.15-5 Logging State Display Dialog Box



3.16 Object Window

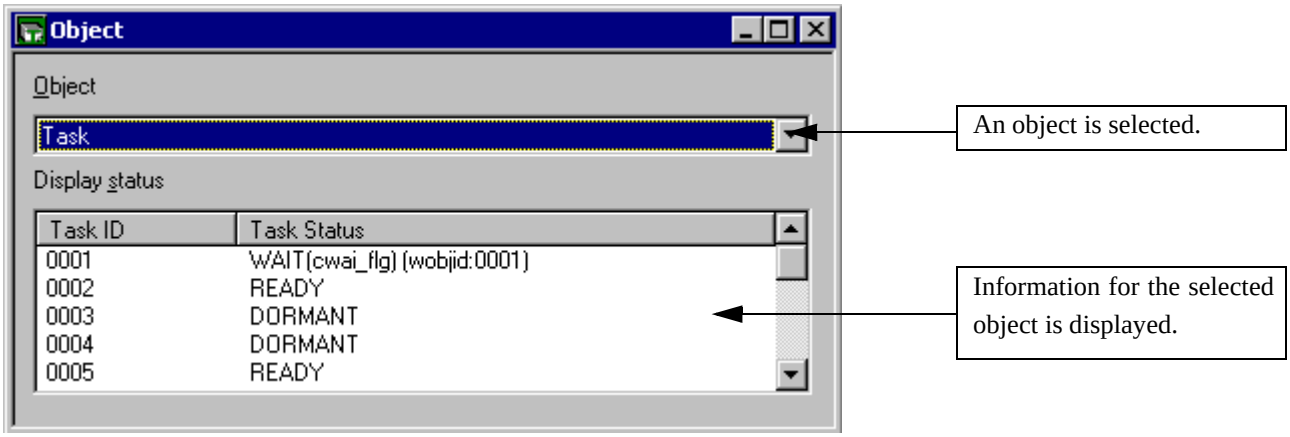
The Object Window displays the object information used by REALOS. This window is displayed only when SOFTUNE WORKBENCH is in the debug session.

■ Object Window

The Object Window displays the object information used by REALOS.

Figure 3.16-1 is an example of the Object Window.

Figure 3.16-1 Example of Object Window



■ Object/State Display

Object	Status display
Task	Task ID, task state
Semaphore	Semaphore ID, count of semaphores
Flag (1-bit flag) Flag (16-word flag)	Flag ID, SET or CLEAR Flag ID, event flag pattern
Mail box	Mail box ID, count of message queues, count of tasks
Memory pool	Count of free blocks, count of tasks
Cyclic handler	Handler ID, active state (residual time till activation)
Alarm handler	Handler ID, set state (residual time till activation)
Ready queue	Priority, count of tasks
Alarm queue	Handler ID
Timer queue	Task ID or cyclic handler ID

■ Shortcut Menu

- Task	Executes task system calls
- Semaphore	Executes semaphore system calls
- Event Flag	Executes event flag system calls
- Mail Box	Executes mail box system calls
- Memory Pool	Executes fixed-length memory pool system calls
- Ready Queue	Executes ready queue system calls
- Cyclic Handler	Executes the system calls related to the cyclic handler.
- Monitoring	Switches whether to enable or disable monitoring
- Properties	Displays object properties
- Close	Closes Object Window

For details on the above items, see Section "4.4.11 Object".

3.17 Coverage Window

The Coverage Window displays the coverage measurement result. This window is displayed only when SOFTUNE WORKBENCH is in the debug session.

■ Coverage Window

Figure 3.17-1 is an example of the Coverage Window. The Coverage Window displays the coverage measurement result at debugging. This window is displayed only in MB2141/MB2147-01 (emulator).

For details on the coverage function, see Section "4.4.12 Coverage".

Figure 3.17-1 Coverage Window (16-Address Unit)

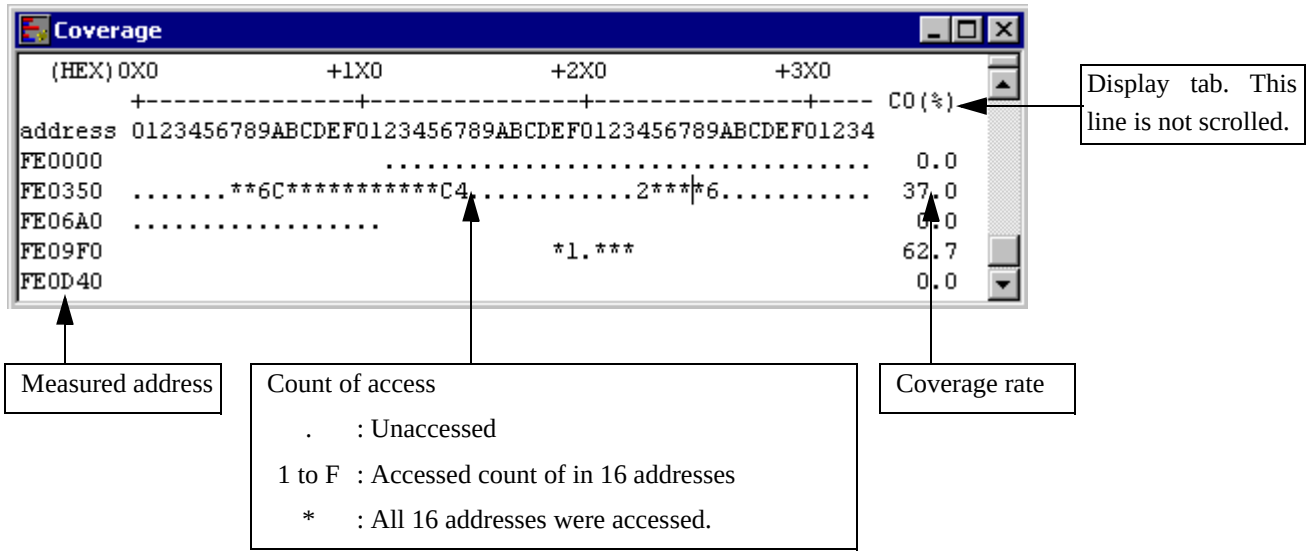
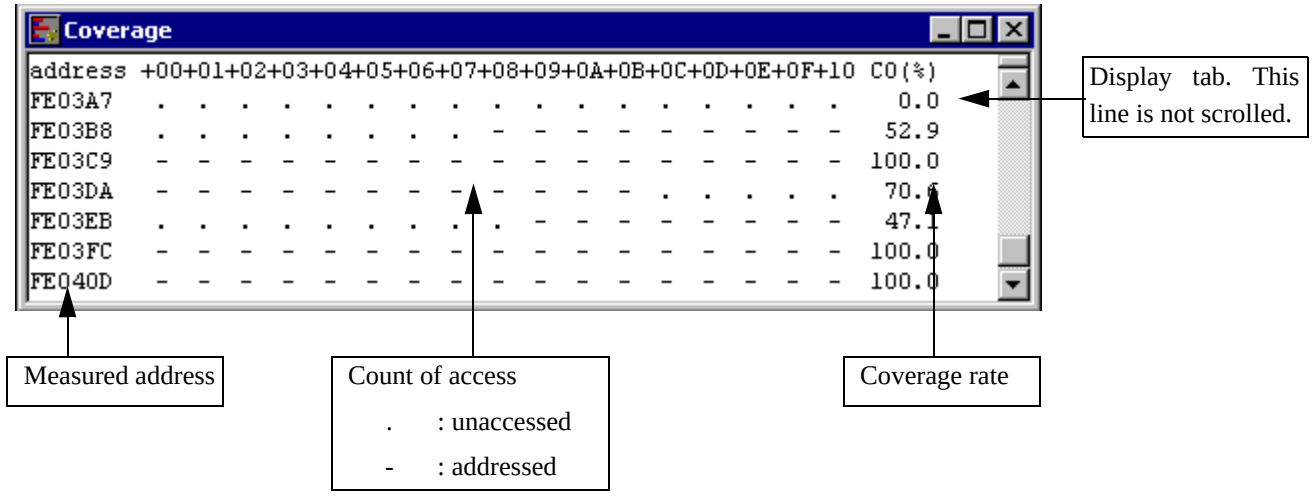


Figure 3.17-2 Coverage Window (1-Address Unit)


Coverage
address +00+01+02+03+04+05+06+07+08+09+0A+0B+0C+0D+0E+0F+10 CO(%)
FE03A7 . . . . . 0.0
FE03B8 . . . . . - - - - - - - - - 52.9
FE03C9 - - - - - - - - - - - - - - - - 100.0
FE03DA - - - - - - - - - - - - 70.6
FE03EB . . . . . - - - - - - - - - 47.1
FE03FC - - - - - - - - - - - - - - - - 100.0
FE040D - - - - - - - - - - - - - - - - 100.0
Display tab. This line is not scrolled.
Measured address
Count of access
. : unaccessed
- : addressed
Coverage rate

■ Function

- Drag and Drop

  By dropping variable name, function mode, label, or address from source window to disassemble window, display will be jumped to a location where an address of dropped character.

■ Shortcut menu

- Update	Updates Coverage Window display.
- Jump	Opens the jump dialog box.
- 16 Address Unit	Displays the coverage measurement result in the unit of 16 addresses.
- 1 Address Unit	Displays the coverage measurement result for each address.
- Set	See Section "4.4.12 Coverage".
- Total Coverage Rate	See Section "4.4.12 Coverage".
- Clear	Clears the coverage measurement buffer.
- Close	Closes the Coverage Window.

3.18 Performance Window

The Performance Window displays the performance measurement result. This window is displayed only when SOFTUNE WORKBENCH is in the debug session.

■ Performance Window

For the performance function, see Section "4.4.13 Performance".

[MB2141]

Examples of the performance window are given in Figure 3.18-1 and Figure 3.18-2. The performance window is used to display the result of performance measurement at debugging. This window is only displayed for the emulator debugger.

Figure 3.18-1 Performance Window (Measurement Count)

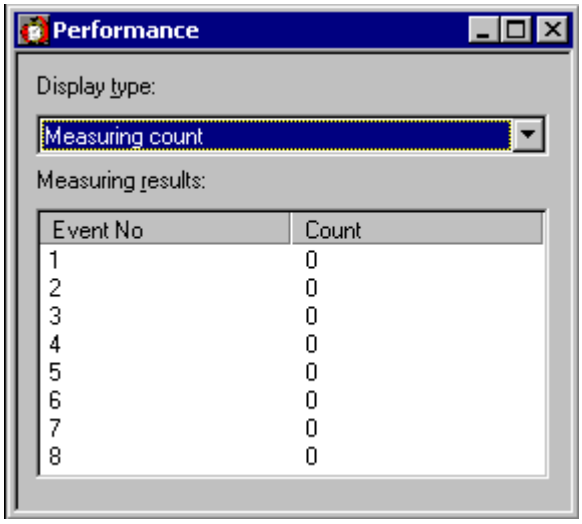
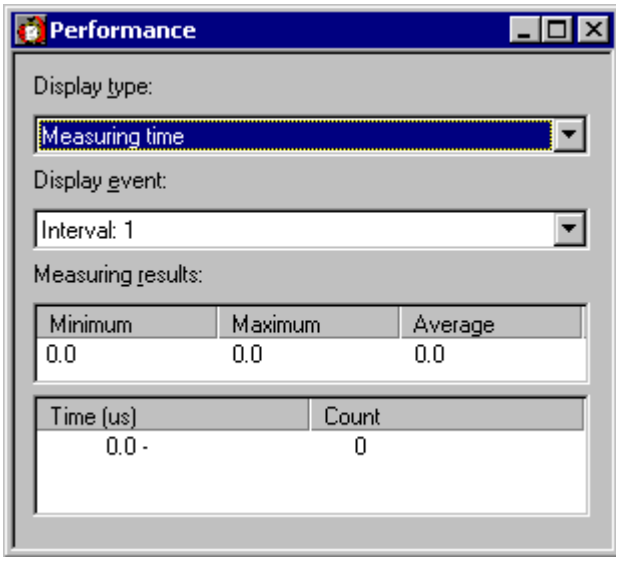


Figure 3.18-2 Performance Window (Measurement Time)



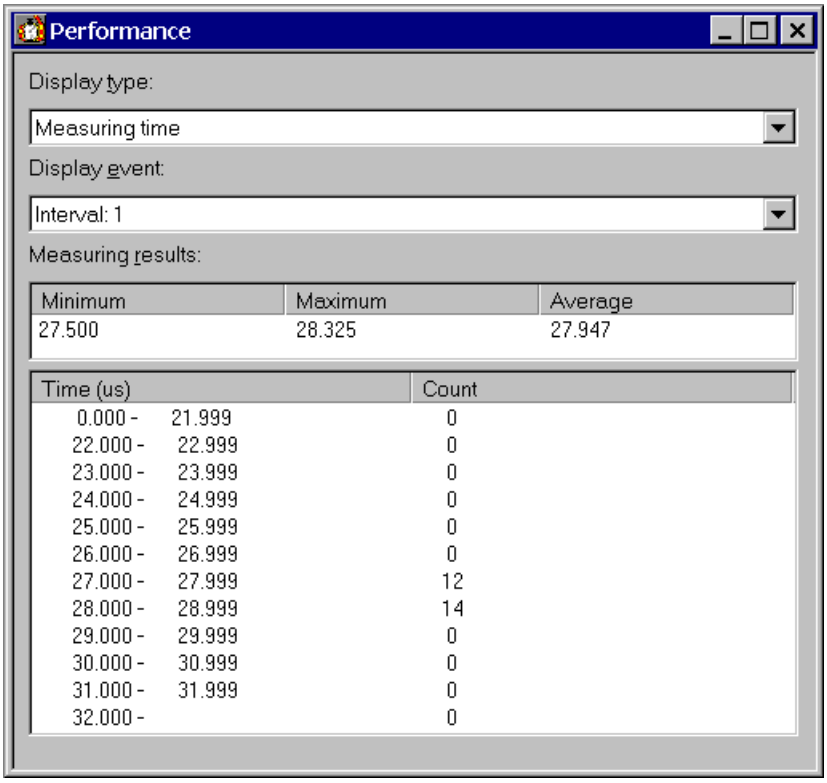
- Display Type — Select measurement time or measurement count as the display type.
- Display Event — A display event is displayed only when measurement time is selected.
  Select a measurement interval.
- Measurement Result — The measurement result is totaled and displayed.
  The minimum measurement result, maximum measurement result, and average measurement result are displayed only when measurement time is selected as the display type.

[MB2147-01]

Figure 3.18-3 is an example of the Performance Window. The Performance Window displays the performance measurement result at debugging.

This window is only displayed for the debug session (emulator) and the performance mode at the event mode.

Figure 3.18-3 Performance Window



- Display Type — Fix in the measurement time.
- Display Event — Select the measurement interval.
- Measurement Result — The measurement result is totaled and displayed.
  The minimum measurement result, maximum measurement result, and average measurement result are displayed only when measurement time is selected as the display type.

■ Shortcut menu

- Update	Updates Performance Window display.
- Set	See Section "4.4.13 Performance".
- Display Range	See Section "4.4.13 Performance".
- Event	See Section "4.6.5 Event".
- Clear	Clears the performance buffer.
- Copy	See Section "4.4.13 Performance".
- Close	Closes the Performance Window.

3.19 Sequence Window

The Sequence Window displays the sequence setting state. This window is displayed only when SOFTUNE WORKBENCH is in the debug session.

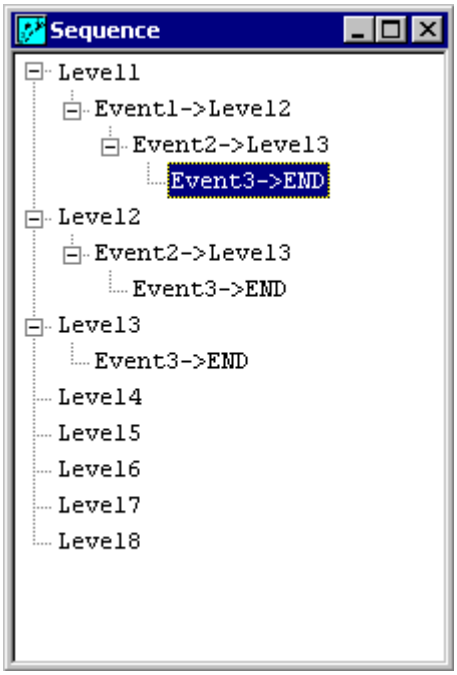
■ Sequence Window

For details on the sequence function, see Section "4.6.6 Sequence".

[MB2141]

An example of the sequence window is shown in Figure 3.19-1. The sequence window is used to display the sequence setting state at debugging. This window is only displayed for the emulator.

Figure 3.19-1 Sequence Window



■ Shortcut menu

- Area	See Section "4.6.6 Sequence".
- Delete	Deletes sequence setting.
- Delay Count	See Section "4.6.6 Sequence".
- Set Latch	See Section "4.6.6 Sequence".
- Display Latch	See Section "4.6.6 Sequence".
- Event	See Section "4.6.5 Event".
- Close	Closes the Sequence Window.

3.20 Real-time Memory Window

This window is displayed only for the debug session. It is used to display the contents of the mirror memory from the specified address in the real-time area.

■ Real-time memory window

An example of the real-time memory window for MB2147-01 is shown in Figure 3.20-1. This window is used to display the contents of the mirror memory specified using [Real-time Area] tab of [Debug Environment] on [Environment] menu at debugging.

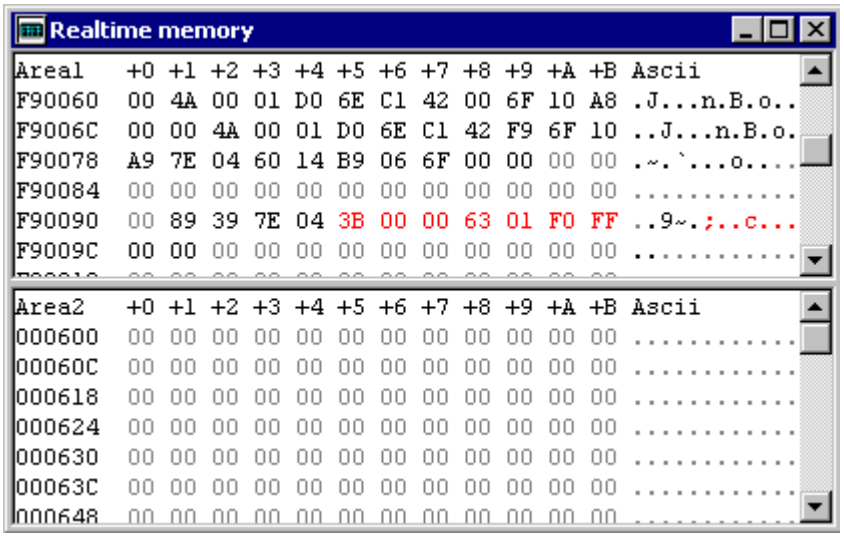
The memory contents cannot be rewritten directly. To change the display area, use [Area] on the shortcut menu.

■ Function

- Drag and Drop

  By dropping variable name, function mode, label, or address from source window to disassemble window, display will be jumped to a location where an address of dropped character.

Figure 3.20-1 Real-time Memory Window



■ Shortcut menu

- Area	Displays data from the beginning of area specified using [Debug Environment] tab of [Real-time Area] on [Environment] menu.
- Area specification	Calls [Real-time Area] tab of [Debug Environment] on [Environment] menu.
- Monitoring	Switches between monitoring enabling and monitoring disabling.
- Close	Closes Real-time memory window

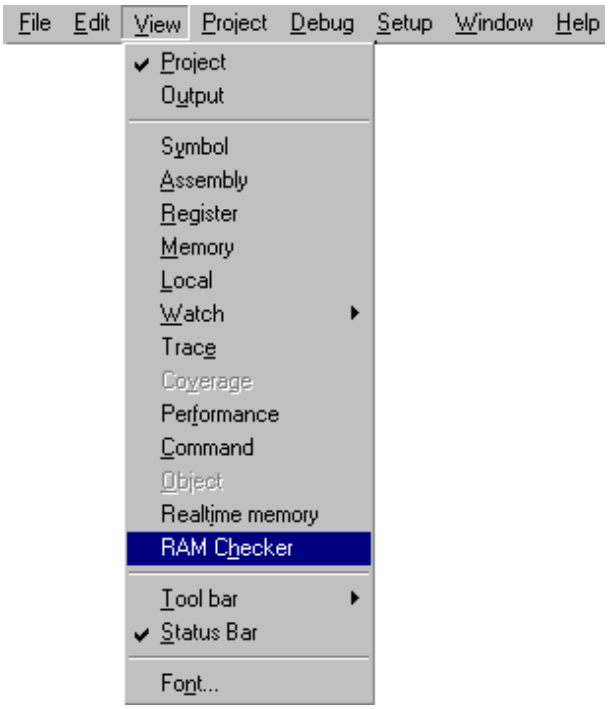
3.21 RAM Checker Window

This window opens only when SOFTUNE WORKBENCH is in the debug session. This window displays logging state and monitoring.

■ RAM Checker Window

Select the main menu [View] - [RAM Checker] to open the RAM Checker window.

Figure 3.21-1 RAM Checker Menu Selection

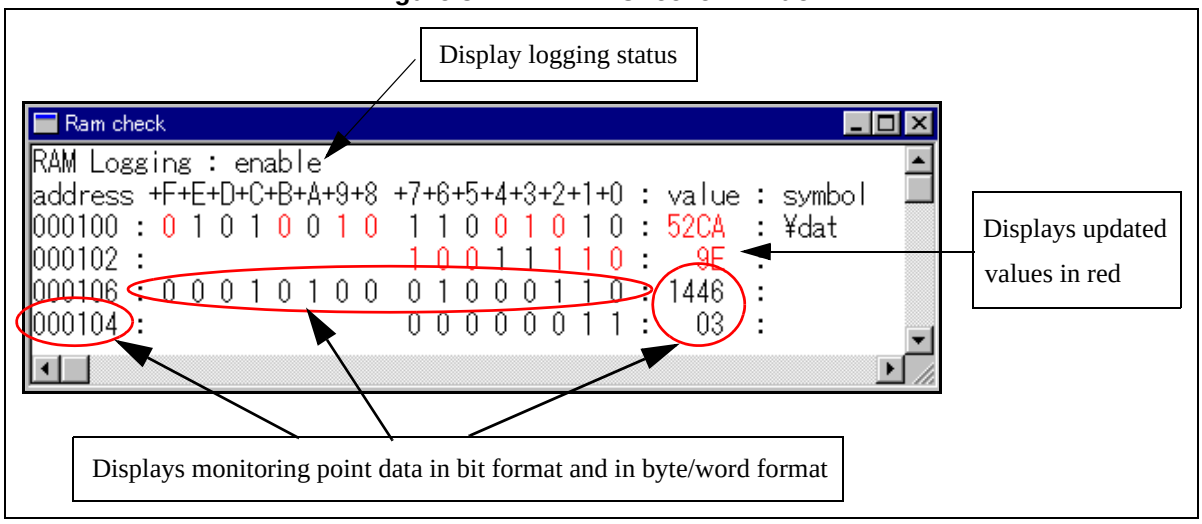


■ Function

- Drag and Drop

  Register the character string dropped on the window, in the RAM Checker window as the monitoring address.

Figure 3.21-2 RAM Checker Window


Display logging status
Ram check
RAM Logging : enable
address +F+E+D+C+B+A+9+8 +7+6+5+4+3+2+1+0 : value : symbol
000100 : 0 1 0 1 0 0 1 0 1 1 0 0 1 0 1 0 : 52CA : ¥dat
000102 : 1 0 0 1 1 1 1 0 : 9E :
000106 : 0 0 0 1 0 1 0 0 0 1 0 0 0 1 1 0 : 1446 :
000104 : 0 0 0 0 0 0 1 1 : 03 :
Displays updated values in red
Displays monitoring point data in bit format and in byte/word format

The RAM Checker window displays the logging status and monitoring.

- If the log status is disabled, monitoring also is disabled.
- Updates monitoring display data every 100 ms.
- Displays data in bit format and byte/word format.
- Displays data that is different from the memory in the previous update in red.
- Cannot edit memory.

Note:

- Monitoring

  When the logging status of the RAM Checker window is enabled, the monitoring display of other windows (memory, watch, etc.) become disabled regardless of their settings.

- Event Mode

  When using the RAM Checker window, set an event mode to single trace mode. If an event mode is set to performance, the RAM Checker window cannot be opened.

  When an event mode is changed to performance mode, the logging state of the RAM Checker window becomes disabled. When an event mode is set to performance while the RAM Checker window is opened, the RAM Checker window automatically closes.

  To set an event mode, open the Setup debug environment dialog box by selecting the menu [Setup] - [Debug Environment] - [Debug Environment...], and then select the event tab.

- Event Functions

  Refer to Sections "2.3.6 Real-time Trace in the USER'S Manual", and Section "4.4.8 Trace" in the Operation Manual for use of event functions.

3.21.1 Setup of the RAM Checker

This section explains setup of the RAM Checker.

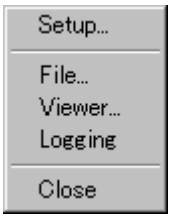
■ Setup

To set the RAM Checker, use the shortcut menu in or input command-line.

This section explains setup by using the shortcut menu. Refer to the Command Reference Manual for setup by using the command-line.

■ Shortcut menu

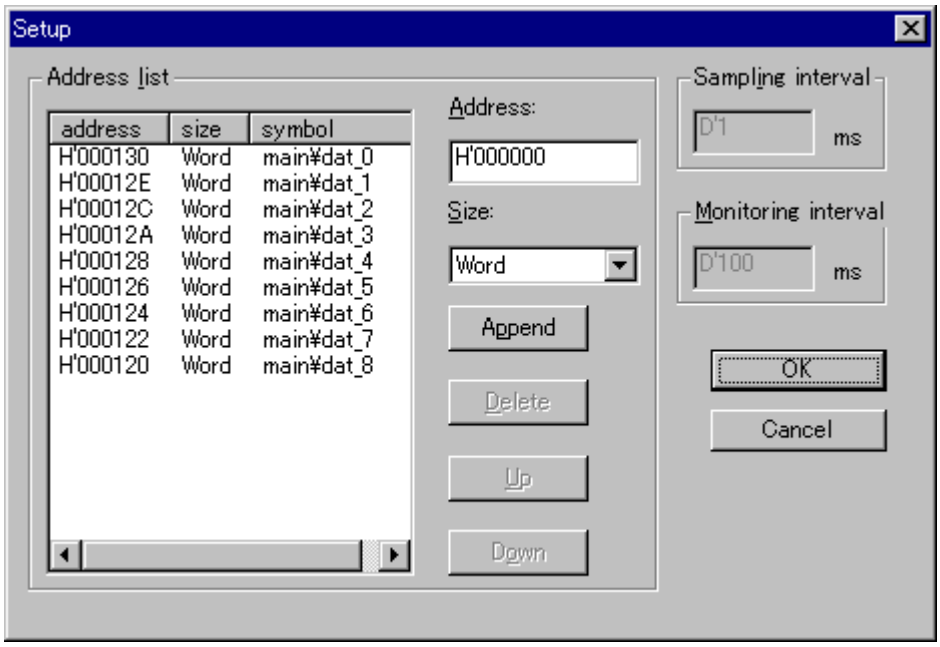
Figure 3.21-3 Shortcut Menu RAM Checker Window



■ Monitoring address

Use the shortcut menu [Setup...] to open the setup dialog box.

Figure 3.21-4 Setup Dialog Box



The RAM Checker window displays the set monitoring addresses in the order of the check address list. It adds setting of new monitoring addresses at the end of the list. The order of the list can be changed.

Sampling intervals (1 ms) and monitoring intervals (100 ms) cannot be changed.

1. Setup of Monitoring Addresses
   - Input monitoring addresses using symbols or addresses.
   - Select either byte or word for data size of monitoring addresses.
   - Press the "Append" button to register them to the list.
   - Press the "OK" button to call the contents of the setup dialog box into the RAM Checker window.

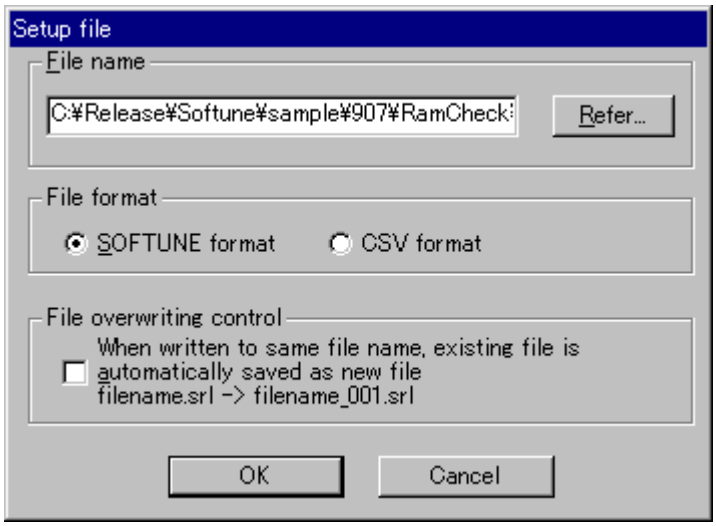
2. Deleting of Monitoring Addresses
   - Select monitoring addresses to be deleted from the list. (Plural choices are possible.)
   - Press the "Delete" button to delete them from the list.
   - Press the "OK" button to call the contents of the setup dialog box into the RAM Checker window.

3. Changing of Display Position
   - Select monitoring addresses to be changed from the list. (Plural choices are possible.)
   - Select a display position using the "Up" or "Down" buttons.
   - Press the "OK" button to call the contents of the setup dialog box into the RAM Checker window.

■ Setup file

Use the shortcut menu [Files...] to open the file setup dialog box.

Figure 3.21-5 Setup File Dialog Box



Specify a log file and data saving format to save the sampling data.

1. File name
   Specify a log filename to be saved. If extensions are omitted, they are automatically applied according type of file format to be selected.
   When the SOFTUNE format is selected, "SRL" is applied.
   When the CSV format is selected, "CSV" is applied.

2. File Format
   Select either SOFTUNE format or CSV format for a file format to be saved.
   - SOFTUNE format: To display in the RAM Checker viewer (recommended)
   - CSV format: To display in other applications than the RAM Checker viewer (*)

   *: The CSV format requires size of data approximately 4 times that of the SOFTUNE format.

3. Overwrite Control

This function obtains data without overwriting the created log files.

When this function is enabled, log files are automatically saved with a different name.

Check off the check box to enable this function.

- Example

If the specified log file (filename.srl) exists, the filename is saved as shown below:

filename.srl → filename_001.srl

If filename_oo1.srl also exists, the filename is saved as shown below:

filename.srl → filename_002.srl

If filename_002.srl§filename_003.srl also exist in the same way, the filenames are saved as shown below:

filename.srl → filename_003.srl

•

•

filename.srl → filename_xxx.srl

- Press the "OK" button to call the contents of the setup dialog box into the RAM Checker window.

Notes:

1. Only internal HDD is supported for the log file storage destination. Network, external HDD and external disk (such as CD, DVD, and MO) are not supported for the log file storage destination.
2. Storing the log file of the RAM Checker requires free disk space of 500MB or greater. When free disk space is less than 500MB, logging stops.

■ Logging

This controls the logging status of the RAM Checker. Check off the shortcut menu to enable this. When the logging is started in this state, the RAM Checker obtains log data and the RAM Checker window displays monitoring.

Figure 3.21-6 Logging Enable Status

■ CPU stop during logging

Power-on debug can be performed during execution of RAM Checker.

Notes:

1. Set the operating lower limit voltage of your microcomputer, as the lower voltage.
2. The emulator outputs undefined values during the period from return from power-on debug to data access. During this period, the Viewer does not display data.

3.21.2 Start the RAM Checker viewer

This section explains about starting of the RAM Checker viewer.

■ Viewer

Use the shortcut menu [Viewer...] to open the Viewer dialog box (Figure 3.21-7). Select the using log file when the RAM Checker viewer is opened, use the button [Execute...] to open the RAM Checker viewer (Figure 3.21-8).

Figure 3.21-7 Viewer Dialog Box

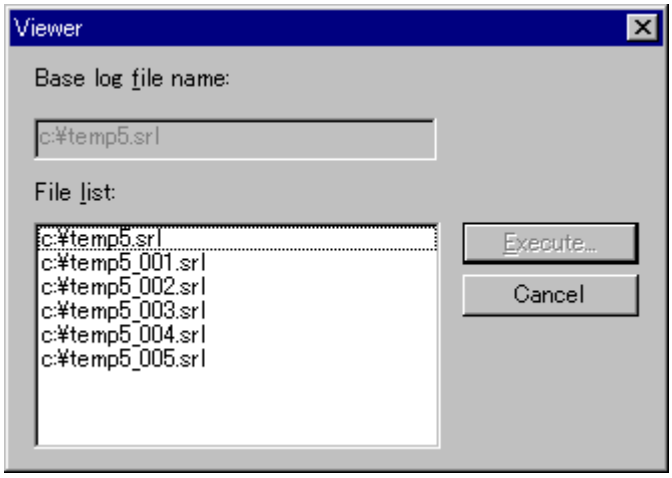
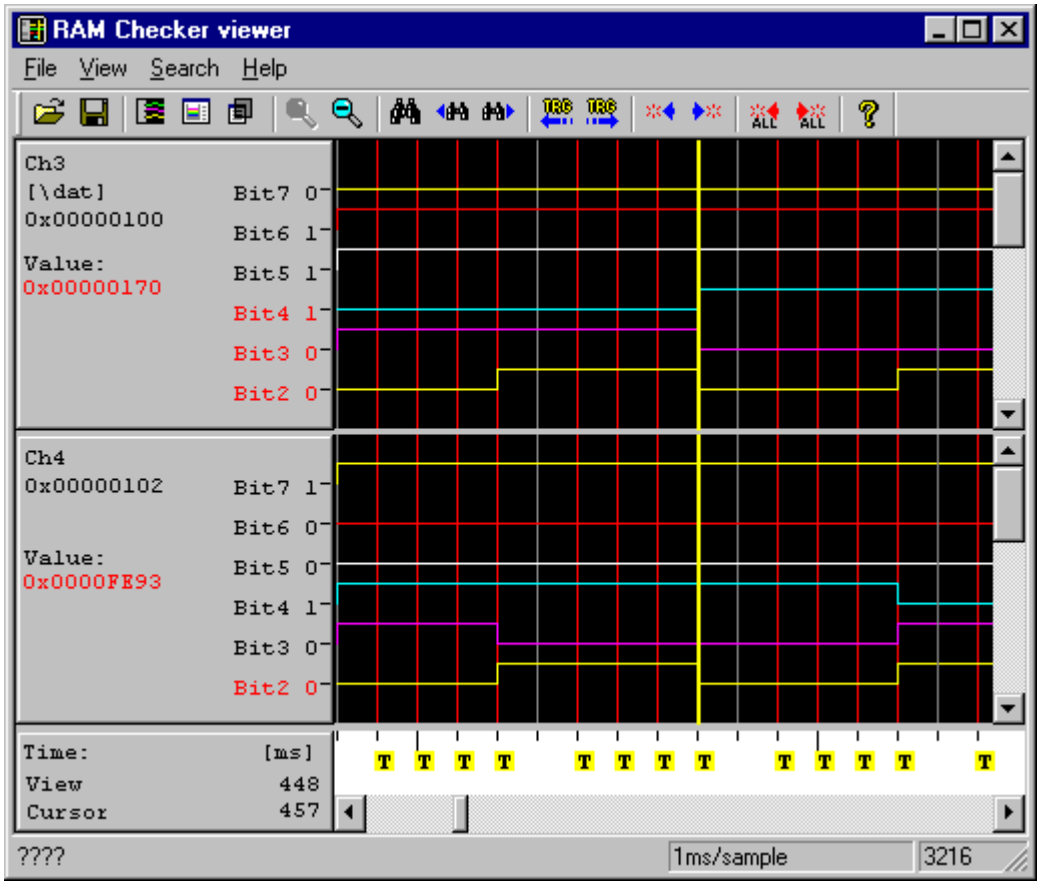


Figure 3.21-8 RAM Checker viewer



Other way, execute the [RAM Checker Viewer] Menu from [SOFTUNE V3] of [FFMC-16 Family SOFTUNE Workbench Tools] of the Start Menu.

Refer to RAM Checker Viewer Manual(FswbRViewE.pdf) and Help for details of the RAM Checker Viewer.

CHAPTER 4
MENUS

This chapter explains the SOFTUNE WORKBENCH menu configuration and the dialog boxes to be started from each menu.

4.1 Menu Configuration (Tree)

4.2 File Menu

4.3 Edit Menu

4.4 View Menu

4.5 Project

4.6 Debug

4.7 Setup

4.8 Window

4.9 Help

4.1 Menu Configuration (Tree)

This section explains the SOFTUNE WORKBENCH menu configuration.

■ Menu configuration

The following menus are displayed in the SOFTUNE WORKBENCH Main Window.

[File]	[Edit]	[View]	[Project]	[Debug]	[Setup]	[Window]	[Help]

The following functions are assigned to each menu item.

- File
  - New
  - Open
  - Close
  - Open Workspace
  - Close Workspace
  - Save
  - Save As
  - Save All
  - Print
  - Recent Text File
  - Recent Project File
  - Exit
- Edit
  - Undo
  - Redo
  - Cut
  - Copy
  - Paste
  - Delete
  - All Select
  - Find
  - Replace
  - Find in Files
  - Jump
  - Bookmark
  - Previous Error
  - Next Error
  - Top of Error

  - Bottom of Error
  - Property
- View
  - Project
  - Output
  - Symbol
  - Assembly
  - Register
  - Memory
  - Local
  - Watch
  - Trace
  - Coverage
  - Performance
  - Command
  - Object
  - Real-time Memory
  - RAM Checker
  - Tool Bar
  - Status Bar
  - Bookmark
  - Fonts
- Project Menu
  - Active Project
  - Add Project
  - Add Member
  - Setup Workspace
  - Setup Project
  - Setting Customized Build
  - Project Dependence
  - Project Configuration
  - Include Dependencies
  - Compile
  - Make
  - Build
  - Stop
- Debug
  - Run
  - Abort

Reset MCU
Break point
Breakpoint Set/Reset
Event
Sequence
Stack
Time Measurement
Call
Clear Call
Vector
Load Target File
Start Debug/Terminate Debug

- Setup Menu

Development
Debug Environment
Memory Map
Tool
Keyboard
Editor
Error
Start Tool

- Window Menu

Cascade
Vertical
Horizon
Split
Arrange Icons
Refresh Window
Refresh All Windows
Close All Windows

- Help Menu

Help Topics
Support Information
About Fsxxxx

4.2 File Menu

File Menu SOFTUNE WORKBENCH has the following functions:

- Workspace and file accessing

- Printing

- End of SOFTUE WORKBENCH

■ Functions related to project and file accessing

The project and file accessing

- New
- Open
- Close
- Open Workspace
- Close Workspace
- Save
- Save As
- Save All
- Recent Text File
- Recent Workspace

■ Other functions

Other functions are as follows.

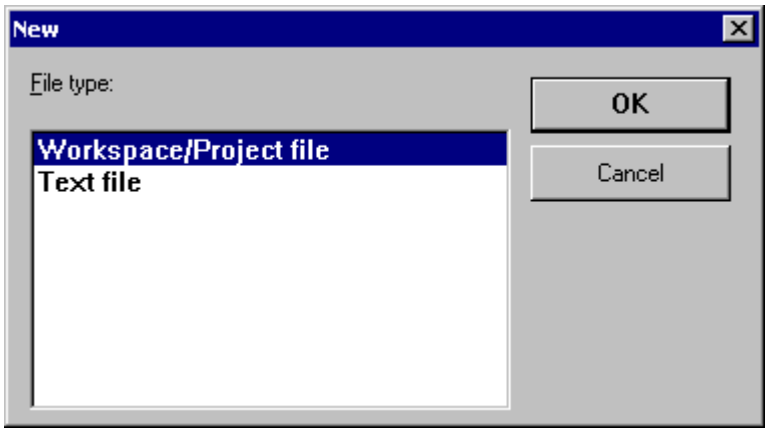
- Print
- Exit

4.2.1 New

"New" creates a workspace/project and file newly. Selecting "New" from the File Menu opens the new creation dialog box asking the operator whether to create a workspace/ project or text file (Figure 4.2-1).

■ New creation dialog box

Figure 4.2-1 New Creation Diary Box

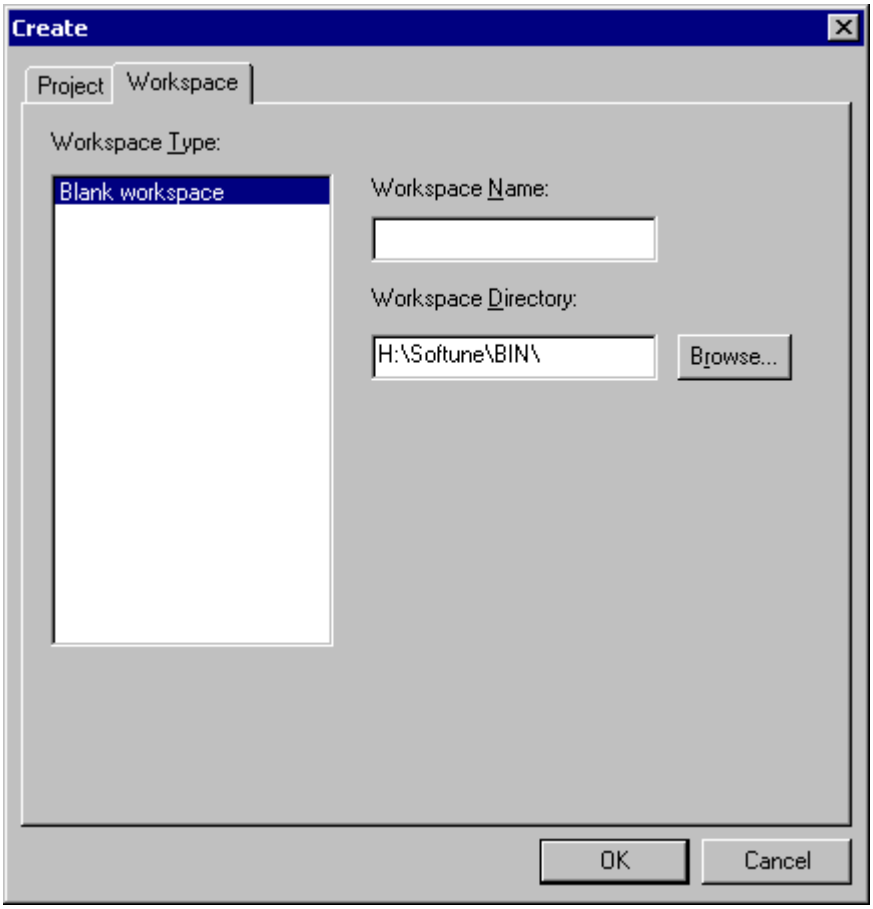


■ Creation of New Workspace

To create a new workspace, select [Workspace/Project File]. When the [New] dialog for project is opened, select the [Workspace] tab (Figure 4.2-2). Set the basic items regarding workspace to be created.

For details about the setting procedure, see Section "2.3 Creating Workspace".

Figure 4.2-2 Project New Dialog (Workspace)

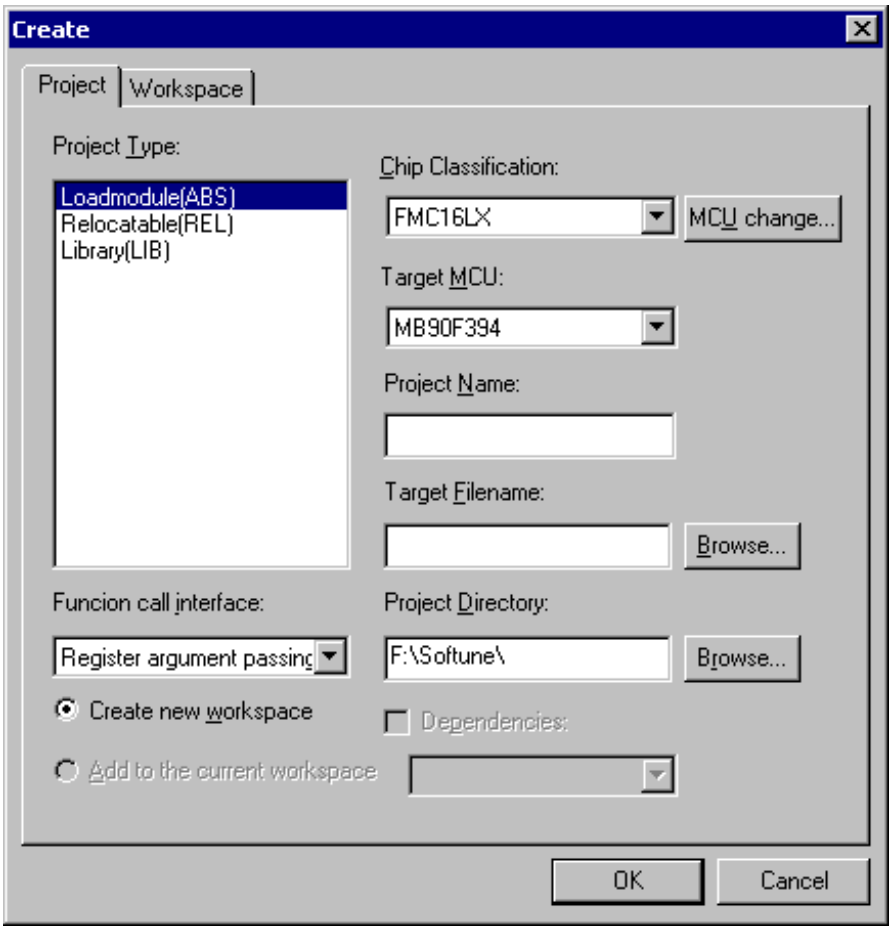


■ Creation of New Project

To create a new project, select [Workspace/Project File]. When the [New] dialog for project is opened, select the [Project] tab (Figure 4.2-3). Set the basic items regarding project to be created.

When creating a new workspace in creating a new project, see Section "2.3 Creating Workspace" and when adding to a existing workspace, see Section "2.4 Storing Project".

Figure 4.2-3 Project New Dialog (Project)



■ Creating source file and document files, etc.

To create a source file and document file, etc., select [Text File] from the new creation dialog box. The editor is started in the status in which a new file is created.

4.2.2 Open

"Open" opens existing files. Selecting "Open" from the File Menu opens the open dialog box asking the file you want to open (Figure 4.2-4).

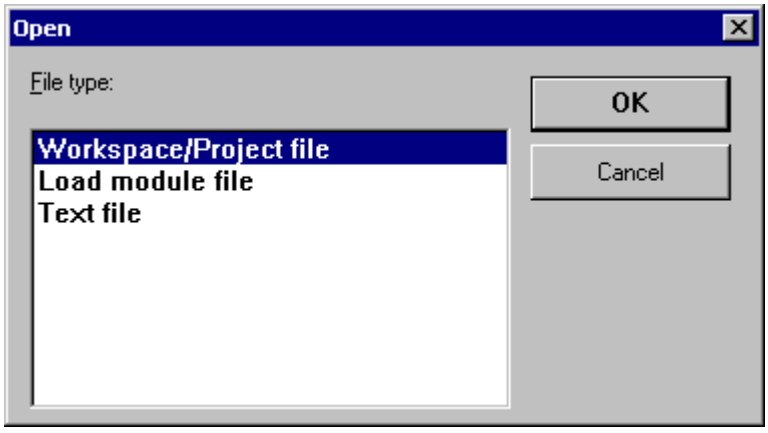
■ Open dialog box

Selecting [Open] from the File Menu opens the open dialog box.

The open dialog box displays a workspace file, project file, load module file, source file, binary file, alias file, coverage file, batch file, and text file.

If SOFTUNE WORKBENCH is not in the debug session, the binary file, alias file, coverage file, and batch file are not displayed.

Figure 4.2-4 Open Dialog Box



■ Opening the load module file

-When the project file has been already opened and SOFTUNE WORKBENCH is in the debug session, the file dialog box for selecting the name of the load module to be debugged opens.

To load debug information only, set a check box for [Debug Information Only].

To set a memory map automatically during simulator debugger, set a check box for [Set Map Automatically]. If this check box is not set, set the memory map before opening the load module file.

-When the project file has been already opened but SOFTUNE WORKBENCH has not been in the debug session the dialog box asking the operator to make SOFTUNE WORKBENCH enter the debug session opens. When the [OK] button is clicked, SOFTUNE WORKBENCH automatically enters the debug session. Clicking the [Cancel] button cancels file open operation.

-When the workspace file is not open, the dialog box for checking whether to create a new workspace/project or whether to open the existing projects opens.

-To create a new workspace/project, click the [Yes] button.

-To read the existing projects, click the [No] button.

-Clicking the [Cancel] button cancels file open operation.

■ Opening the binary file

Specify the start address for binary data read in [Start Address].

■ Opening the batch file

Setting [Run in Icon State] starts batch processing in the icon state.

4.2.3 Close

"Close" closes open files and windows.

■ Close

[Close] closes the files open in the Edit Window or debug windows (e.g., Register Window).

-Edit Window

When an unsaved modified file is in the Edit Window, the dialog box for asking the operator whether to save the file opens.

-[Yes] button ... Saves the file using the existing file name.

-[No] button ... Does not save the file and closes the Edit Window.

-[Cancel] button ... Processing is continued with the file being edited and opened.

-Debug windows (e.g., Register Window)

SOFTUNE WORKBENCH closes debug windows immediately.

4.2.4 Open Workspace

"Open Workspace" opens the existing workspace file.

■ Opening the existing workspace file

The file dialog box for selecting the file you want to open opens.

-When the existing workspace file has been already opened and SOFTUNE WORKBENCH is in the debug session, the dialog box for asking the operator whether to terminate debugging opens.

-[OK] button ... Terminates the debugger and continues workspace open processing.

-[Cancel] button ... Cancels workspace open processing.

-When the existing workspace file has been already opened but SOFTUNE WORKBENCH is not in the debug session, the file dialog box for selecting a workspace file opens. When a workspace file name is specified, SOFTUNE WORKBENCH closes the currently open workspace file and then opens the specified file. If the open registered file is not saved although it is being edited, the dialog box for asking the operator whether to save the file opens.

-[Yes] button ... Saves the file using the existing file name and continues processing.

-[No] button ... Continues processing without saving the file.

-[Cancel] button ... Cancels workspace open processing.

-When the existing workspace file is not open, the file dialog box for selecting a project file opens. When a workspace file name is specified, SOFTUNE WORKBENCH opens the specified file. When a workspace file is opened, the used window when the file was saved is redisplayed.

-[Cancel] button ... Processing is continued with the file being edited opened.

The project file can be opened instead of the workspace file. For details, see Section "2.3 Creating Workspace".

4.2.5 Close Workspace

"Close Workspace" closes the currently open workspace file.

■ When the currently open workspace file is not edited at all

SOFTUNE WORKBENCH tries to close the file immediately.

When the workspace file is closed, the dialog box for asking the operator whether to save the current workspace information opens.

-[Yes] button ... Saves the current workspace information and closes the workspace file.

-[No] button ... Closes the project file without saving the current workspace information.

-[Cancel] button ... Cancels workspace file close processing.

In the following cases, however, the above dialog box does not open:

-The workspace file is not modified.

-When the workspace file is closed, inquiry for save is not set.

■ When the currently open project file is being edited

The dialog box for asking the operator whether to save the current workspace information opens. When the [Yes] or [No] button is clicked to continue processing, the dialog box for asking the operator whether to save the file being edited subsequently opens.

-[Yes] button ... Saves the file being edited and then closes the workspace file.

-[No] button ... Closes the file being edited without saving the file being edited.

-[Cancel] button ... Workspace is closed with the file being edited opened.

4.2.6 Save

"Save" saves the currently open file using the existing file name.

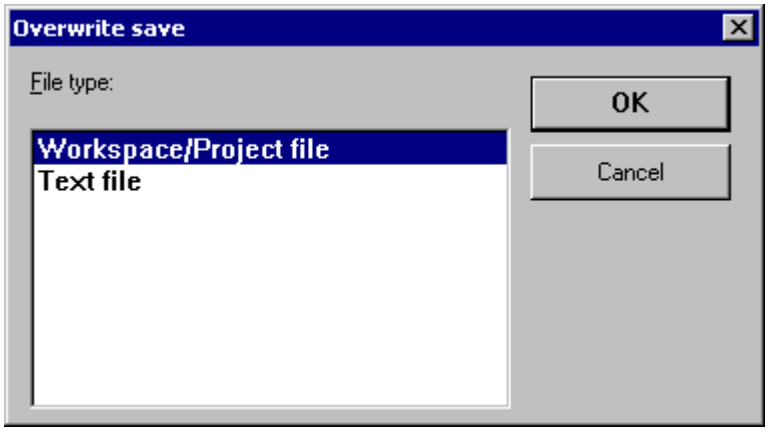
■ Save dialog box

The save dialog box is used to save the file using the existing file name.

The text file, project file, binary file, alias file, setup file, or coverage file can be selected from this dialog box as the type of file.

If SOFTUNE WORKBENCH is not in the debug session, the binary file, alias file, setup file, and coverage file are not displayed.

Figure 4.2-5 Save Dialog Box



4.2.7 Save As

"Save As" stores the previously saved file under a new file name.

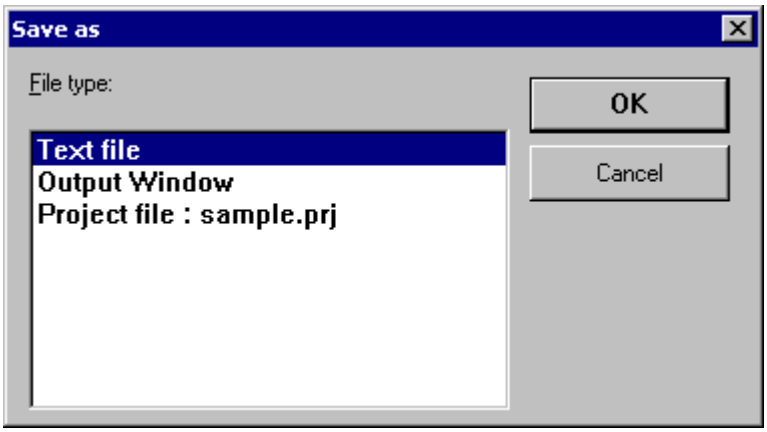
■ Save As

[Save As] stores the previously saved file under a new file name.

The text file, output window, project file, binary file, alias file, setup file, or coverage file can be selected as the type of file.

If SOFTUNE WORKBENCH is not in the debug session, the binary file, alias file, setup file, and coverage file are not displayed.

Figure 4.2-6 Save as Dialog Box



■ Saving the project file

Save the active project file.

Select the format for saving the project file from the [File Type] combo box.

For the project formats, refer to Section "1.2 Management Function for Project" of SOFTUNE WORKBENCH User's Manual.

If a Workspace project format (*.prj) is selected, it saves all project information in workspace project format (compatible to workspace).

When the project file opened in old project format is overwritten in workspace project format, conversion is performed and restrictions on old project files are cleared. Subsequent saving is performed in workspace project format.

If an Old project format (*.prj) is selected, it saves information on active project configurations in old project format (incompatible to workspace).

The project file opened in workspace project format cannot be overwritten in old project format. In this case, save the file as a different name.

■ Saving the binary file

Specify a save start address in [Start Address]. When the binary file has been already loaded, the start address at that time is set.

Specify a save end address in [End Address]. When the binary file has been already loaded, the end address at that time is set.

Note:

If the project file in a different directory from the original one is saved, the members described in a relative path may not be referred. In this case, move the members with the original directory configuration kept being unchanged.

4.2.8 Save All

"Save All" saves all the files currently being edited.

■ Files to be saved

The files to be saved include text files such as the source file open in the edit window. Data in the workspace, project file and output window is not saved.

Even the text file open in the edit window is not saved if it is not edited.

When SOFTUNE WORKBENCH is in the debug session, the binary, alias, setup, or coverage file that was edited even once is to be saved.

■ Processing a newly created text

If there is a newly created text that has not been saved yet, the file dialog box for asking the operator to enter the name of the file to be saved opens. Specify a file name from this dialog box.

4.2.9 Print

"Print" prints file data.

■ Data to be printed

[Print] prints data in the file currently open in the edit window; it cannot print other window (file) data.

Print output window data in any of the following two procedures:

-Copy output window data to the edit window in the following procedure, then select [Print] from the [File] menu:

1.Select all output window data.

2.Click the right button of the mouse to display the menu, then select [Copy].

3.Open the edit window, then select [Paste] from the [Edit] menu.

-Save output window data once, then print it.

For how to save output window data, see Section "4.2.7 Save As".

Open the saved file with the edit window, then select [Print] from the [File] menu.

4.2.10 Recent Text File/Recent Workspace File

Of the source files opened in the Source Window, "Recent Text File" stores up to five most recently opened source files. Of the used project files, "Recent Project File" stores up to five most recently opened workspace files.

■ Recent Text File

Of the source files opened in the Source Window, [Recent Text File] stores up to five most recently opened source files. It enables to open the desired text file immediately by selecting it.

The files in the Edit Window are not stored here.

■ Recent Workspace File

Of the used project files, [Recent Workspace File] stores up to five most recently opened project files. It enables you to open the desired project file immediately by selecting it.

Note:

If a text or project file has been already deleted or moved to another directory, it cannot be opened even if it is listed here.

4.2.11 Exit

"Exit" terminates SOFTUNE WORKBENCH.

■ When a file is being edited

The dialog box for asking the operator whether to save the file being edited opens.

[Yes] ... Saves the file being edited and terminates SOFTUNE WORKBENCH.

[No] ... Terminates SOFTUNE WORKBENCH without saving the file being edited.

[Cancel] ... Does not terminate SOFTUNE WORKBENCH.

■ When the workspace is open

The dialog box for asking the operator whether to save the current workspace information opens.

[Yes] ... Saves the current workspace information in the file.

[No] ... Does not save the current workspace information in the file.

[Cancel] ... Does not terminate SOFTUNE WORKBENCH.

If no file is being edited, SOFTUNE WORKBENCH terminates immediately. If there is a file being edited, processing explained in "When a file is being edited" is performed.

■ When SOFTUNE WORKBENCH is in the debug session

As in processing explained in "When the workspace is open", the dialog box for asking the operator whether to save the current workspace information opens. Clicking [Yes] or [No] from this dialog box continues processing and automatically terminates the debug session.

If no file is being edited, SOFTUNE WORKBENCH terminates immediately. If there is a file being edited, processing explained in "When a file is being edited" is performed.

4.3 Edit Menu

The edit menu provides such functions as file editing, find string, and error jump.

■ File editing function

The file editing function includes the following:

- Undo
- Redo
- Cut
- Copy
- Paste
- Delete
- All Select

■ Character string search function

The character string search function includes the following:

- Find
- Replace
- Find in Files

■ Jump function

Enables to move to any place in a window quickly:

- Jump

■ Bookmark function

Enables to setup a bookmark or to jump to the bookmark:

- Bookmark

■ Error jump function

The error jump function includes the following:

- Previous Error
- Next Error
- Top of Error
- Bottom of Error

■ Property function

To display the information of activated window:

- Property

4.3.1 Undo, Redo

"Undo" cancels the immediately preceding editing and undoes the status before the editing.

■ Target

This function is available only for edits for the edit window. It can also continuously cancel or undo a "series of actions" for successive addition or deletion of characters and lines.

Note:

This function is valid only for the edit window. Note that the value changed in the Memory or Register Window cannot be undone when SOFTUNE WORKBENCH is in the debug session.

[Undo], [Redo] function may not always work for a large amount of edits.

4.3.2 Cut, Copy, Paste, Delete

"Cut" cuts the selected character string and "Copy" copies it to the clipboard. "Delete" deletes the selected character string. "Paste" inserts the copied character string into the cursor position or replaces it with the selected character string.

■ Cut

"CUT" cuts the character string selected from the edit window and moves it to the clipboard. This command can be executed only in the edit window.

The character string cut here can be pasted later.

■ Copy

"COPY" copies the character string selected from the edit window or the Symbol Window, Assembly Window, Register Window, Memory Window, Local Symbol Window, Watch Window, Trace Window, Command Window or Coverage Window (valid in debug session) to the clipboard.

The character string copied here can be pasted later.

■ Paste

"PASTE" inserts the cut character string or the character string copied to the clipboard into the current cursor position. If the selected character string is in the edit window, this command also replaces it with the character string in the clipboard.

■ Delete

"DELETE" deletes the character string selected from the edit window. This command can be executed only in the edit window.

The deleted character string cannot be pasted later.

Table 4.3-1 Relationship between Edit Functions and Windows

Window	Cut	Copy	Paste	Delete
Edit Window	○	○	○	○
Project Window	X	X	X	X
Output Window	X	○	X	X
Symbol Window	X	○	X	X
Assembly Window	X	○	X	X
Register Window	X	○	X	X
Memory Window	X	△ *1	X	X
Local Symbol Window	X	△ *2	X	X
Watch Window	X	△ *2	X	X
Trace Window	X	△ *1	X	X
Command Window	X	○	X	X
Coverage Window	X	△ *1	X	X
Performance Window	X	X	X	X
Sequence Window	X	X	X	X
Object Window	X	X	X	X

*1: Lines other than the column title can be copied.

*2: Only the symbol name can be copied.

4.3.3 Select All

"Select All" selects all the character strings displayed in a window.

■ Target

This function operates only in the edit and output windows.

4.3.4 Find/Replace

"Find" searches the text file displayed in the edit window for any character string. "Replace" replaces the found character string with the specified character string.

■ Target

"FIND" and "REPLACE" can be executed only for the text files displayed in the edit window; they cannot be executed in the output window and others.

■ Find

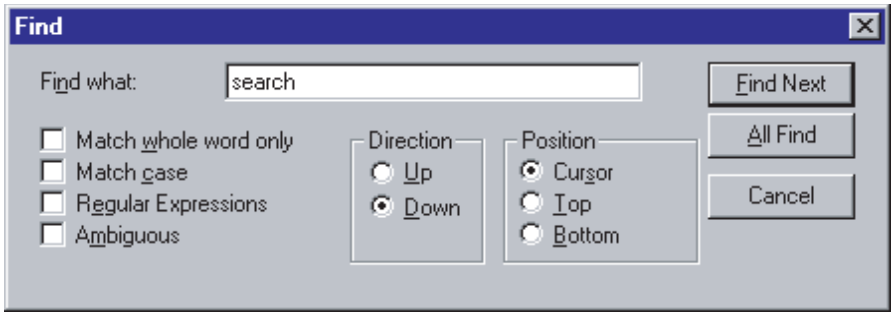
When the find dialog box shown in Figure 4.3-1 opens, specify the character string you want to find from this dialog box. Character string search conditions (Table 4.3-2) can also be specified from this dialog box.

The found character string is displayed in reverse video.

The find dialog box is not automatically closed irrespective of whether the character string is found. For this reason, when search terminates, click the [Cancel] button to close the find dialog box.

When SOFTUNE WORKBENCH is in the debug session, the Source Window can also be searched.

Figure 4.3-1 Find Dialog Box



■ Replace

When the replace dialog box shown in Figure 4.3-2 opens, specify a search character string and a replacing character string. In this case, character string search conditions (Table 4.3-2) can be also specified from this dialog box.

The found character string is displayed in reverse video. Clicking the [Replace] button replaces the specified search character string with the specified replacing character string. Clicking the [All Replace] button replaces all the character strings found before search terminates.

The replace dialog box is not automatically closed irrespective of whether the found character string was replaced. For this reason, when search terminates, click the [Cancel] button to close the replace dialog box.

Figure 4.3-2 Replace Dialog Box

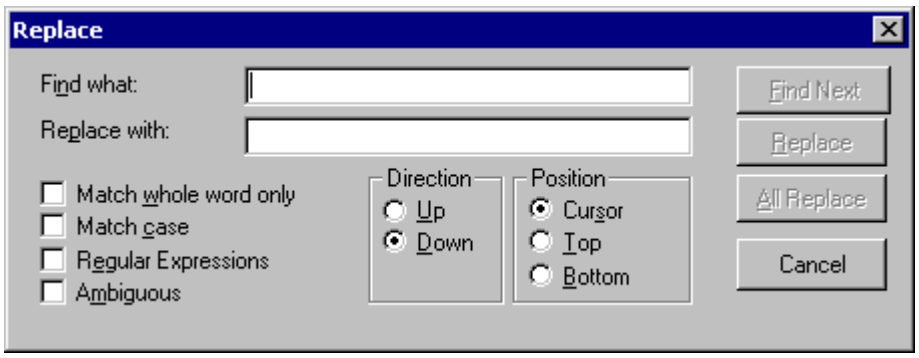


Table 4.3-2 Character String Search Conditions

Condition	Explanation
Match whole word only	Does not search partial character strings of words. Effective in searching for independent words that may become parts of other words such as "able".
Match case	Distinguishes uppercase characters from lowercase characters or vice versa.
Regular Expression	Specifies a search character string in regular expression. Enables use of a wildcard, etc.
Direction	Specifies a search direction (Up or Down). Up searches the file upward. Down searches the file downward.
Position	Specifies a search start position (cursor, top, bottom).

4.3.5 Find in Files

This section explains how to search the specified file for the specified character string.

■ File search

SOFTUNE WORKBENCH searches the specified file for the specified character string and displays the search result in the output window.

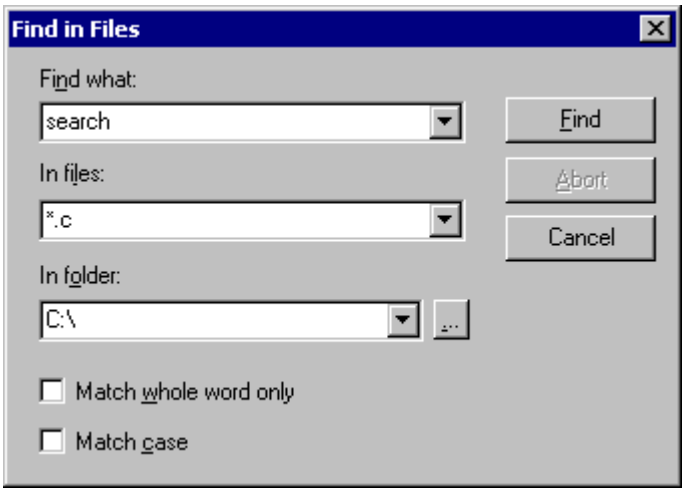
SOFTUNE WORKBENCH can search several files at a time.

A line can be immediately edited by double-clicking the mouse on the line in the output window.

■ Search procedure

1. Select the [Edit]-[Find in Files] menu.
   Dialog box Figure 4.3-3 opens.
2. Specify a character string.
   The dialog box showing the character strings in the clipboard opens.
3. Specify the file to be searched.
   A wild card can also be used to specify the file to be searched.
4. Specify the directory to be searched.
   Click the button to the right of the specified field to open the directory search dialog box.
5. Specify [Match whole word only] and [Match case] as required.
6. Click the [Find] button.
   The search result appears in the output window in real time.
7. To abort a search, click [Abort] button.
   The file search is aborted.
8. When search terminates, click the [Cancel] button.

Figure 4.3-3 File Search Dialog Box



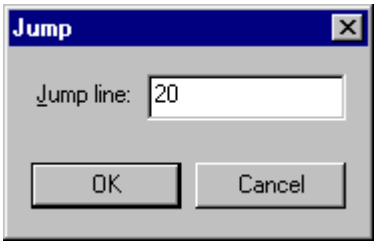
4.3.6 Jump

"Jump" moves the cursor to any line in the text file being edited. When SOFTUNE WORKBENCH is in the debug session, this function (command) specifies the display start position of a source line, disassemble, memory, or trace, coverage.

■ Edit window

When the jump dialog box shown in Figure 4.3-4 opens, specify a jump destination line number. When the edit window is active, the cursor in the edit window jumps to the specified line irrespective of whether SOFTUNE WORKBENCH is in the debug session.

Figure 4.3-4 Jump Dialog Box (Edit)



■ Debug session

If a window other than the edit window becomes active when SOFTUNE WORKBENCH is in the debug session, the dialog box shown in Figure 4.3-5 opens. Select [TYPE], then specify [POSITION] in the selected format.

If [Address] is specified in [Type], [window] at source, memory or disassemble window can be used to specify the window where jump is implemented.

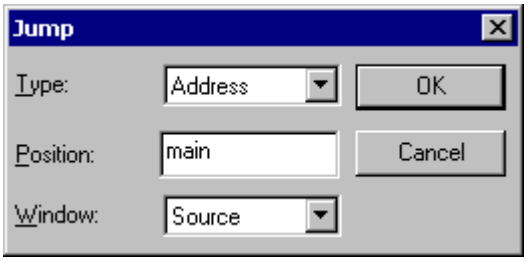
The following can be selected as [TYPE].

- Line number
- Address
- Frame

The following can be selected as [WINDOW].

- Source Window
- Memory Window
- Disassemble Window
- Realtime Memory Window

Figure 4.3-5 Jump Dialog Box (Debug)



4.3.7 Bookmark

It is convenience in setting a bookmark for referring the location specified for source, memory and edit windows often. Once a bookmark is set, there will be a mark on specified location. This enables to jump to the specified location by using menu or key operation. Bookmark will be validated until it is canceled.

■ Bookmark

Bookmark is a function to specify the location in window and enable to jump to the location by using menu or key operation.

■ Compliant window

Bookmark is valid in following windows.

- Source window
- Memory window
- Edit window

■ Bookmark menu

Menu related to bookmark is located at [Edit]-[Bookmark] in a menu.

Sub menu consists of 7 types as follows. Menu becomes valid when it is in debug session or each window is activated.

- Bookmark
- Setup/cancel bookmark
- Next bookmark
- Previous bookmark
- Next bookmark in the current window
- Previous bookmark in the current window
- Clear all bookmarks in the current window

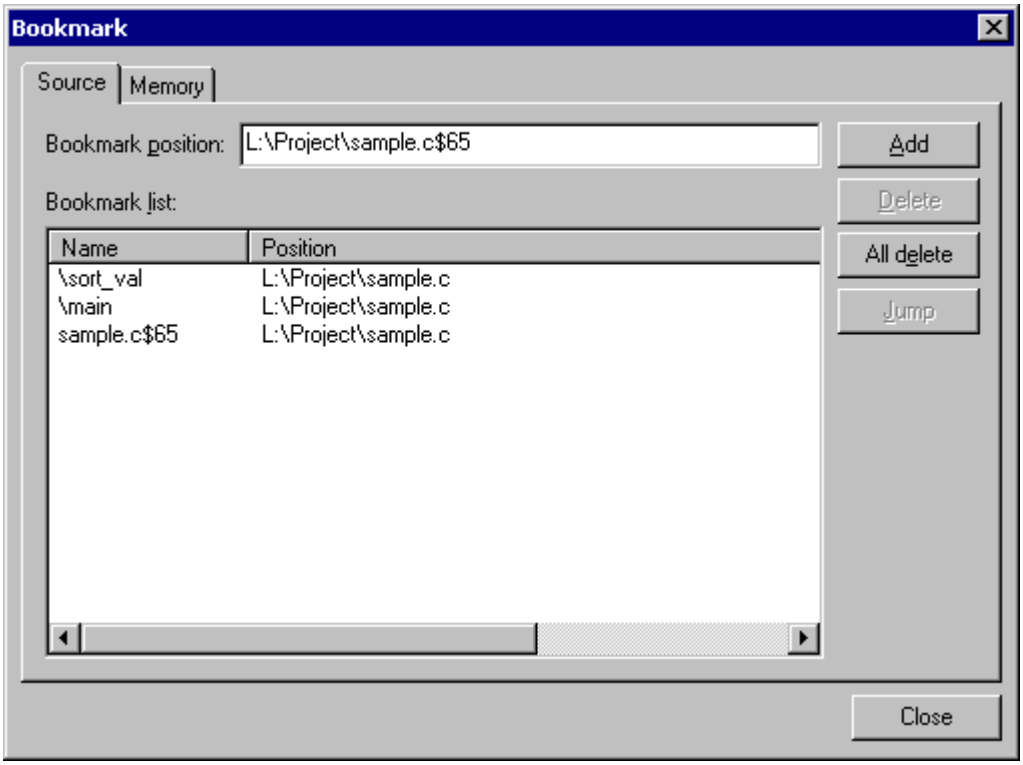
4.3.7.1 Bookmark - Source window

This section explains the operation when source window is activated.

■ To display a dialog "Bookmark"

By selecting [Edit]-[Bookmark]-[Bookmark], dialog as shown in Figure 4.3-6 will be opened. This dialog is to indicate the list of bookmark set in source window.

Figure 4.3-6 Bookmark dialog (Source window)



- Bookmark position
  - This is a field to specify the position to set a bookmark. Entering relative path, function name or label name into the field can specify position of bookmark.
- Bookmark list
  - The list of bookmark, which is currently set in a source window, will be displayed.

    Name : Name of bookmark will be displayed. Name of bookmark will be appeared as line number or symbol name.

    Position : File name, of which bookmark is set, will be displayed.

    Line : Line number, where bookmark is set, will be displayed.

    Symbol : Symbol, which is allocated to bookmark name, will be displayed. If a symbol is allocated, Bookmark position will be updated based on debug information although the line number of symbol changes along with the change in file. Please refer to [Note] for allocatable symbol.

- [Add] button
  - Set a bookmark to a location specified by [Bookmark position.
- [Delete] button
  - Delete selected bookmark from a list.
- [All delete] button
  - Delete all bookmarks from a bookmark list.
- [Jump] button
  - Move a start position for displaying a source window to location of bookmark selected in bookmark list.

■ Setup or cancel of a bookmark

1) Please put a cursor to line where you want to set or cancel a bookmark.
2) Please execute any of following operations.
   - Select [Edit]-[Bookmark]-[Setup/cancel bookmark]
   - Click on [Setup/cancel bookmark] at toolbar [Find]
   - Add or delete bookmark to/from [Source] tab in dialog [bookmark]

■ Move the caret to the next bookmark

1) Please make sure that there is a cursor in a source window.
2) Please execute any of following operations.
   - Select [Edit]-[Bookmark]-[Subsequent bookmark]
   - Click on [Subsequent bookmark] at toolbar [Find]
3) If no subsequent bookmark is exist on a window, proceed to the subsequent bookmark located on the next file.

■ Move the caret to the previous bookmark

1) Please make sure that there is a cursor in a source window.
2) Please execute any of following operations.
   - Select [Edit]-[Bookmark]-[Previous bookmark]
   - Click on [Previous bookmark] at toolbar [Find]
3) If no previous bookmark is existed on a window, proceed to previous bookmark located on the last file.

■ Move the caret to the next bookmark in the current window

1) Please make sure that there is a cursor in a source window.
2) Please execute any of following operations.
   - Select [Edit]-[Bookmark]-[Next bookmark in the current window]
   - Click on [Next bookmark in the current window] at toolbar [Find]

■ Move the caret to the previous bookmark in the current window

1) Please make sure that there is a cursor in a source window.
2) Please execute any of following operations.
   - Select [Edit]-[Bookmark]-[Previous bookmark in the current window]
   - Click on [Previous bookmark in the current window] at toolbar [Find]

■ Clear all bookmarks in the current window

1) Please make sure that there is a cursor in a source window.

2) Please execute any of following operations.

- Select [Edit]-[Bookmark]-[Clear all bookmarks in the current window]
- Click on [Clear all bookmarks in the current window] at toolbar [Find]
- Click on [All delete] at tab [Source] in a dialog [Bookmark]

Note:

Symbol, which can not be allocated to the name of bookmark, will be member of automatic variables, structure, union, and class.
If some symbols with the same name are defined, other symbol than assumed one may be used. To specify a symbol with the same name, please determine the name of module and function name first, then specify a symbol.

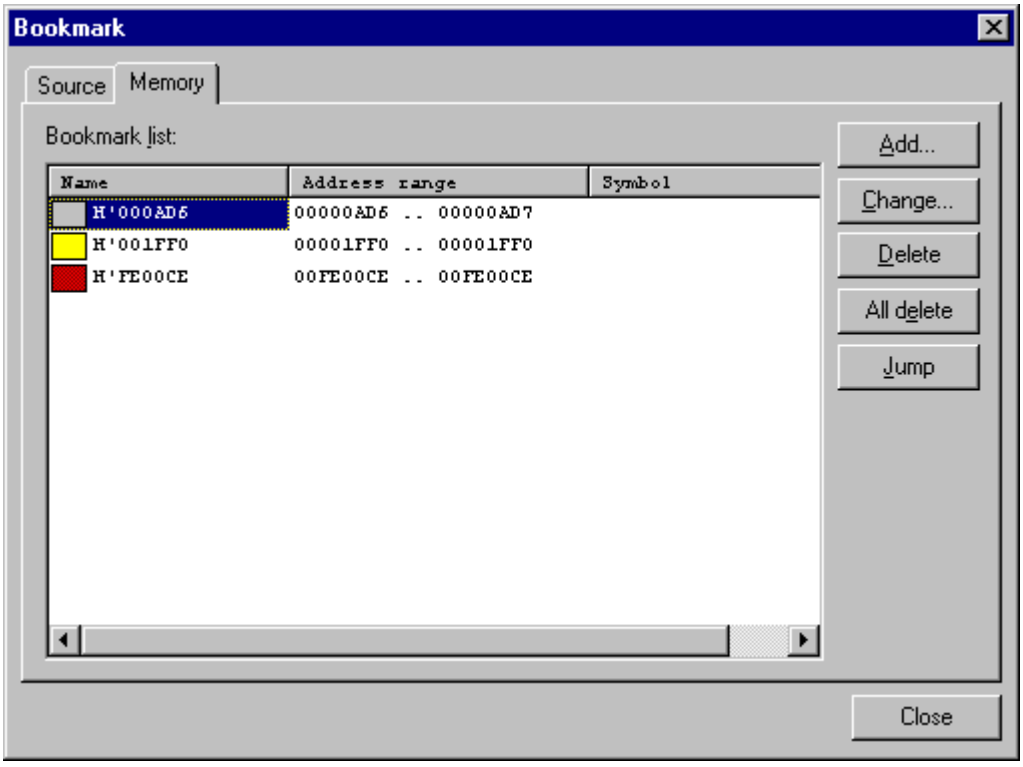
4.3.7.2 Bookmark - Memory window

This section explains the operation when memory window is activated.

■ To display a dialog "Bookmark"

By selecting [Edit] -[Bookmark] - [Bookmark], dialog as shown in Figure 4.3-7 will be opened. This dialog is to indicate the list of bookmark set in memory window.

Figure 4.3-7 Bookmark dialog (Memory window)



- Bookmark list
  - The list of a bookmark, which is currently set in memory window, will be displayed.

    Name :Name of a bookmark and color of marking will be displayed. Address or symbol name will be name of bookmark.

    Address range :Range of an address where bookmark is set is displayed.

    Symbol :Symbol allocated to a bookmark will be displayed. If a symbol is allocated, location of bookmark will be updated based on debug information although the line number of symbol changes along with the change in file. Please refer to [Note] for allocatable symbol.
- [Add] button
  - Display dialog [Add a bookmark] to add a bookmark. For details, refer to "4.4.5 Memory".
- [Change] button
  - Display [Add a bookmark] dialog and change the setup of selected bookmark in a bookmark list.

- [Delete] button
  - Delete a bookmark selected from the list of a bookmark.
- [All Delete] button
  - Delete all bookmarks from the list of a bookmark.
- [Jump] button
  - Move a start position for displaying a memory window to position of bookmark selected in bookmark list.

■ Setup or cancel of a bookmark

1) Please select a range of an address where you want to set or to cancel a bookmark
2) Please execute any of following operations.
   - Select [Edit]-[Bookmark]-[Setup/cancel bookmark]
   - Click on [Setup/cancel bookmark] mark at toolbar [Find]
   - Setup a book mark with short cut menu [Add a bookmark] (For details, refer to "4.4.5 Memory")
   - Add or delete a book mark at tab [Memory] in the Bookmark dialog

■ Move the caret to the next bookmark in the current window

1) Please make sure that there is a cursor in a memory window.
2) Please execute any of following operations.
   - Select [Edit]-[Bookmark]-[Next bookmark in the current window]
   - Click on [Next bookmark in the current window] at toolbar [Find]

■ Move the caret to the previous bookmark in the current window

1) Please make sure that there is a cursor in a memory window.
2) Please execute any of following operations.
   - Select [Edit]-[Bookmark]-[Previous bookmark in the current window]
   - Click on [Previous bookmark in the current window] at toolbar [Find]

■ Clear all bookmarks in the current window

1) Please make sure that there is a cursor in a memory window.
2) Please execute any of following operations.
   - Select [Edit]-[Bookmark]-[Clear all bookmarks in the current window]
   - Click on [Clear all bookmarks in the current window] at toolbar [Find]
   - Click on [All delete] at tab [Memory] in the Bookmark dialog

---

Note:

Symbol, which can not be allocated to the name of bookmark, will be member of automatic variables, structure, union, and class.
If some symbols with the same name are defined, other symbol than assumed one may be used. To specify a symbol with the same name, please determine the name of module and function name first, then specify a symbol.

4.3.7.3 Bookmark - Edit window

This section explains the operation when edit window is activated

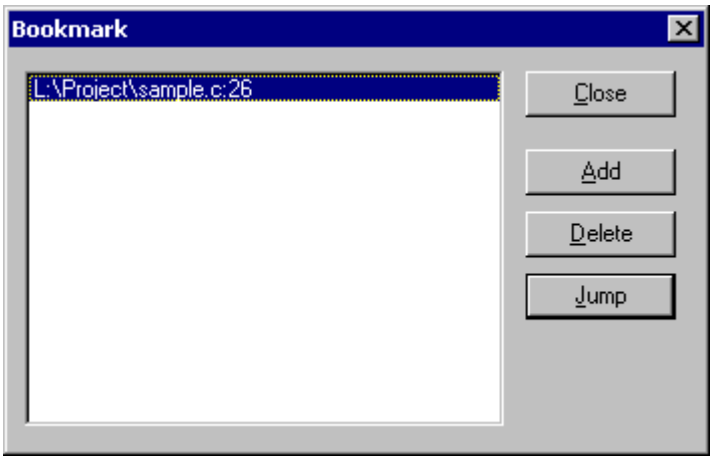
■ To display a dialog "Bookmark"

By selecting one of following menus, dialog as shown in Figure 4.3-8 will be opened. This dialog is to display the list of bookmark that is set to an edit window.

1. [Edit]-[Bookmark]-[Bookmark]

2. Short cut menu of Edit window [Bookmark]

Figure 4.3-8 Bookmark dialog (Edit window)



- Bookmark list
  - The list of a bookmark (file name, line number), which is set currently to an edit window, will be displayed.
- [Add] button
  - Setup of a bookmark to location of an edit window where cursor is put.
- [Delete] button
  - Delete selected bookmark from a list. When more than one bookmarks are selected, delete all items.
- [Jump] button
  - Move a start position for displaying a edit window to location of bookmark selected in bookmark list.

■ Setup or cancel of a bookmark

1) Please move a cursor to line where you want to set or cancel a bookmark.

2) Please execute any of following operations.

- Select [Edit]-[Bookmark]-[Setup/cancel bookmark]
- Click on [Setup/cancel bookmark] mark at toolbar [Find]
- Click [Add] or [Delete] button at the Bookmark dialog.

■ Move the caret to the next bookmark in the current window

1) Please make sure that there is a cursor in a edit window.

2) Please execute any of following operations.

- Select [Edit]-[Bookmark]-[Next bookmark in the current window]
- Click on [Next bookmark in the current window] at toolbar [Find]

■ Move the caret to the previous bookmark in the current window

1) Please make sure that there is a cursor in a edit window.

2) Please execute any of following operations.

- Select [Edit]-Bookmark]-[Previous bookmark in the current window]
- Click on [Previous bookmark in the current window] at toolbar [Find]

■ Clear all bookmarks in the current window

1) Please make sure that there is a cursor in a edit window.

2) Please execute any of following operations.

- Select [Edit]-[Bookmark]-[Clear all bookmarks in the current window]
- Click on [Clear all bookmarks in the current window] at toolbar [Find]

4.3.8 Previous error, Next error, Top of Error, Bottom of Error

"Previous error", " Next error", "Top of Error" and "Bottom of Error" moves the cursor in the edit window to the source line where a make, build, compilation, or assembler error occurred.

■ Previous error

The cursor jumps to the error line immediately before the cursor position in the output window. If there is no error-line before the cursor position, the cursor loops back and jumps to the bottom error line.

■ Next error

The cursor jumps to the error line immediately after the cursor position in the output window. If there is no error line after the cursor position, the cursor loops back and jumps to the top error line.

■ Top of error

The cursor jumps to the top error line displayed in the output window.

■ Bottom of error

The cursor jumps to the bottom error line displayed in the output window.

4.3.9 Property

"Property" displays information for the specified file.

■ Property

There are six windows below.

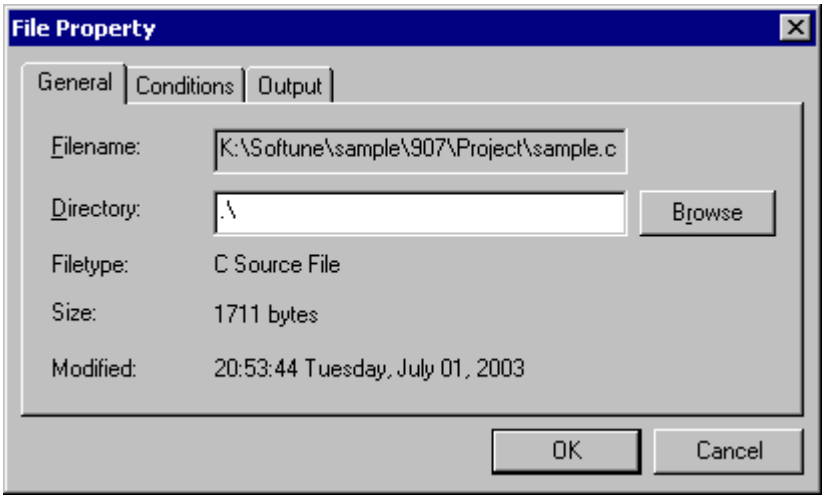
- Project window
- Edit window
- Source window
- Symbol window
- Local window
- Watch Window

4.3.9.1 Property-Project Window

This section explains the property in the Project Window.

■ Property: General

Figure 4.3-9 Property-Project Window Dialog Box (General 1)



Display the file information, module information, and symbol information. The information to be displayed depends on the currently active window.

There are six windows below.

The full-path to the selected file is displayed.

-File name

The full-path to the selected file is displayed.

-Directory

The directory of the selected file is displayed. If editing is enabled, the file to be referred can be changed. If given in a relative path, the directory is described in a relative path in the project file.

-File type

The type of file managed in the project is displayed.

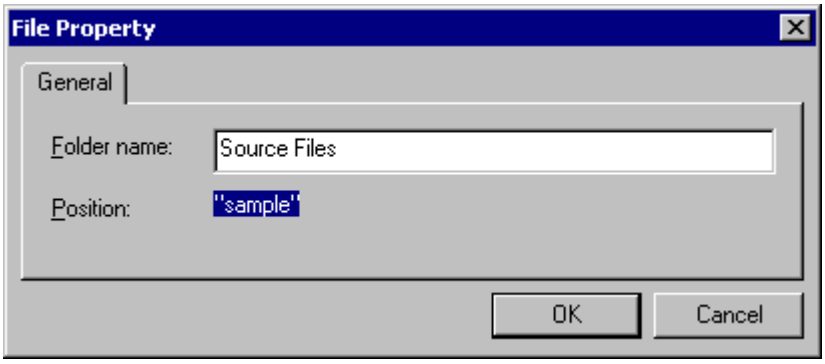
-Size

Displays the file size.

-Modified

Displays the date and time of the last file update.

Figure 4.3-10 Property-project (General 2)



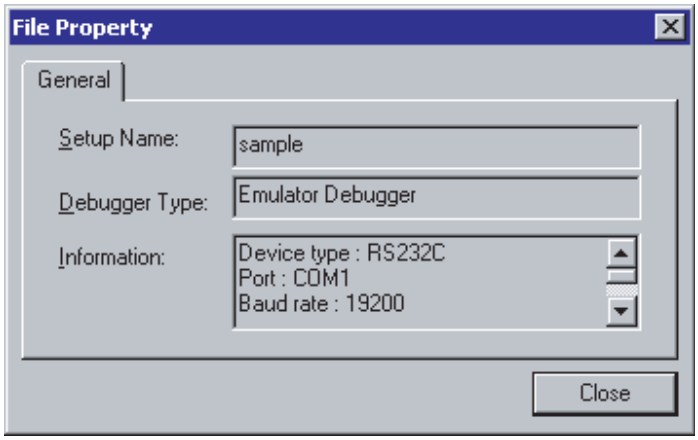
-Folder name

The name of the selected folder is displayed. If editing is enabled, folder name can be changed.

-Place

The place of the selected folder is displayed.

Figure 4.3-11 Property-Project Window (General 3)



-Setup name

The selected setup name is displayed.

-Debugger type

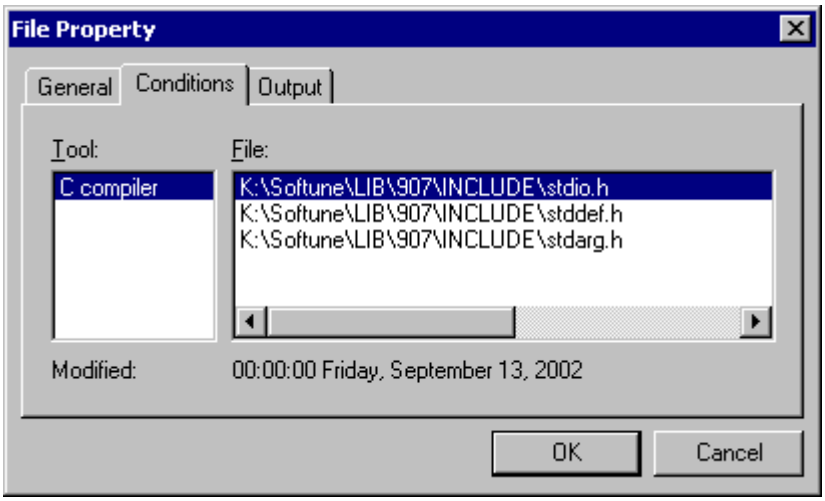
The Debugger type set in Debugger setup is displayed.

-Information

The device type, host, port, baud rate, batch file before performing load, batch file after performing load, etc., are displayed. The displayed information depends on the device type, etc.

■ Property: Conditions

Figure 4.3-12 Property-Project Window (Dependence)



-Tool

Displays the language tool to be executed.

-File

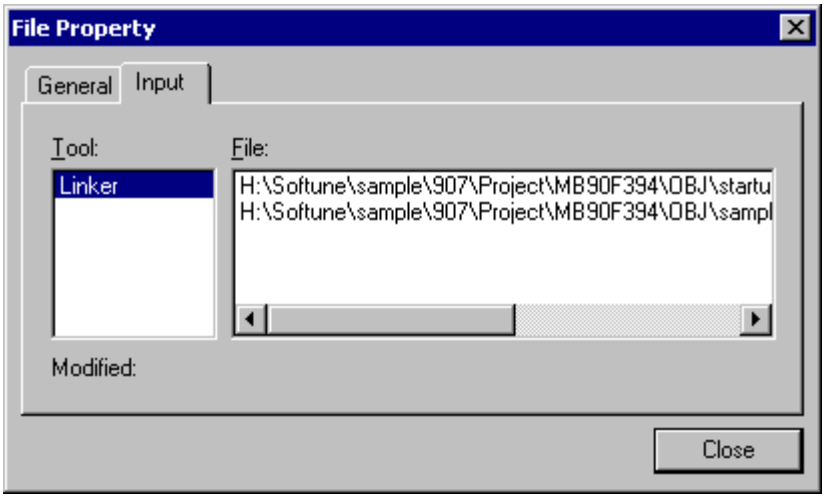
Displays a dependency file list. Selecting a file name from the list displayed the update date and time of the file.

-Updating date

Displays the date and time of the last file update.

■ Property: Input

Figure 4.3-13 Property-Project Window (Input)



-Tool

Displays the language tool to be executed.

-File

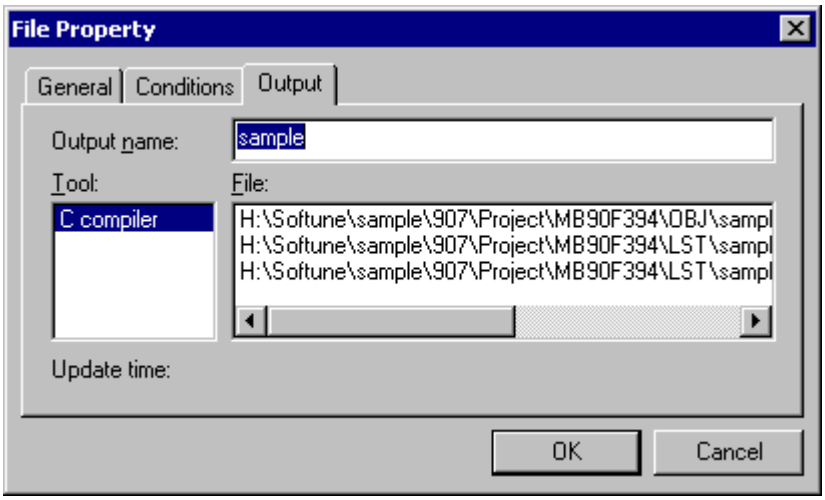
The file used in creating a target file is displayed in order.

-Updating date

Displays the date and time of the last file update.

■ Property: Output

Figure 4.3-14 Property-Project Window (Output)



-Output name

Base name of output file

-Tool

Displays the language tool to be executed.

-File

Output file list created by the language tool.

-Updating data

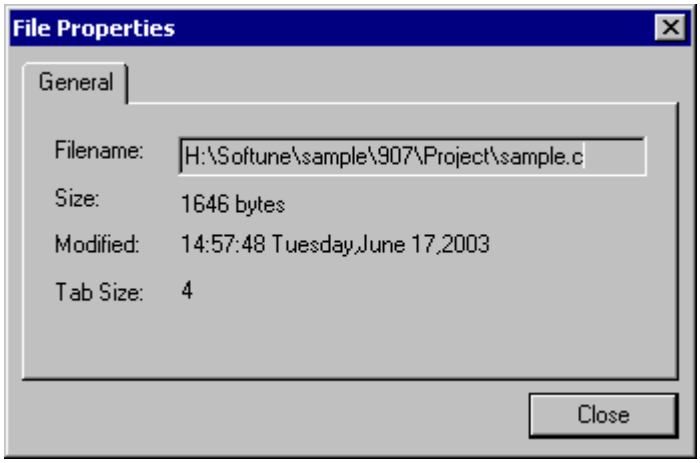
Displays the date and time of the last file update.

4.3.9.2 Property-Edit Window

This section explains the property in the Edit window.

■ Property: General

Figure 4.3-15 Property-Edit Window (General)



-File name

The full-path to the selected file is displayed.

-Size

Displays the file size.

-Updating date

Displays the date and time of the last file update.

-Tab size

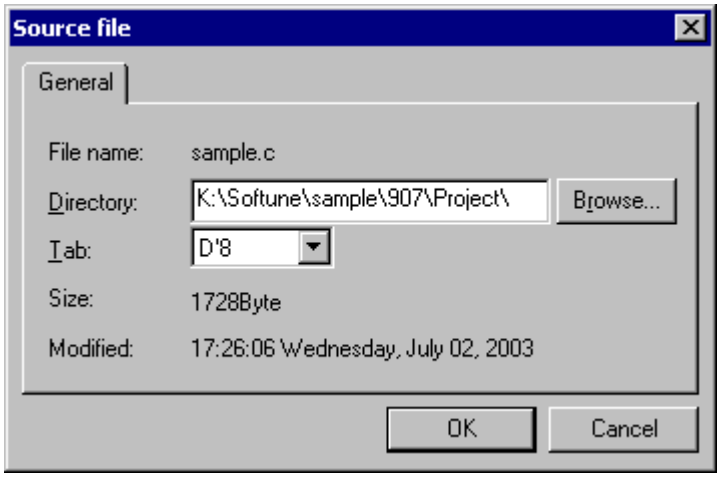
The size of tab code is displayed

4.3.9.3 Property-Source Window

This section explains the property in the Source window.

■ Property: General

Figure 4.3-16 Property-Source Window (General)



-File name

The full-path to the selected file is displayed.

-Directory

Displays the directory in which the source file exists (valid only when Source Window information is displayed). When changed, the directory is searched and the found source file is redisplayed.

-Tab

Displays the tab size. It is possible to change.

-Size

Displays the file size.

-Updating date

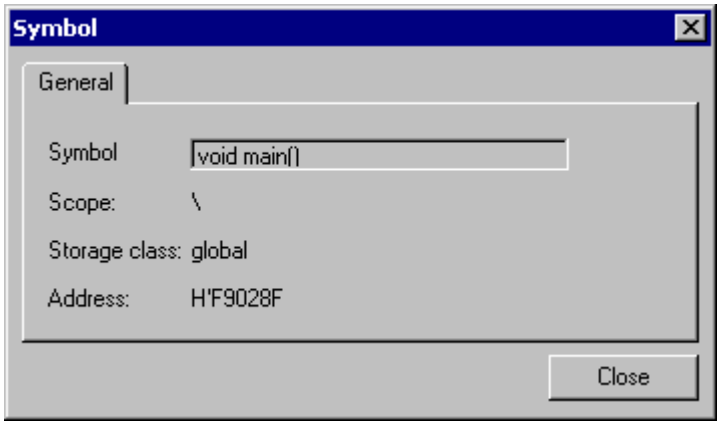
Displays the date and time of the last file update.

4.3.9.4 Property-Symbol Window

This section explains the property in the Symbol window.

■ Property: General

Figure 4.3-17 Property-Symbol Window (General)



-Symbol name

The name of the selected symbol is displayed.

-Scope

The scope position of the selected symbol is displayed.

-Storage class

The storage class of the selected symbol is displayed.

-Address

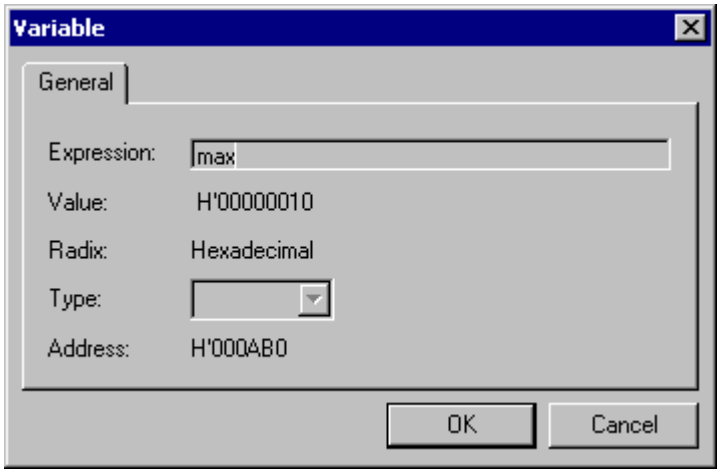
The address of the selected symbol is displayed.

4.3.9.5 Property-Local Window

This section explains the property in the Local window.

■ Property: General

Figure 4.3-18 Property-Local Window (General)



-Expression

The selected item is displayed.

-Value

The value of the selected expression is displayed.

-Base

The base in which the value is given is displayed.

-Type

The type of the selected expression is displayed.

-Address

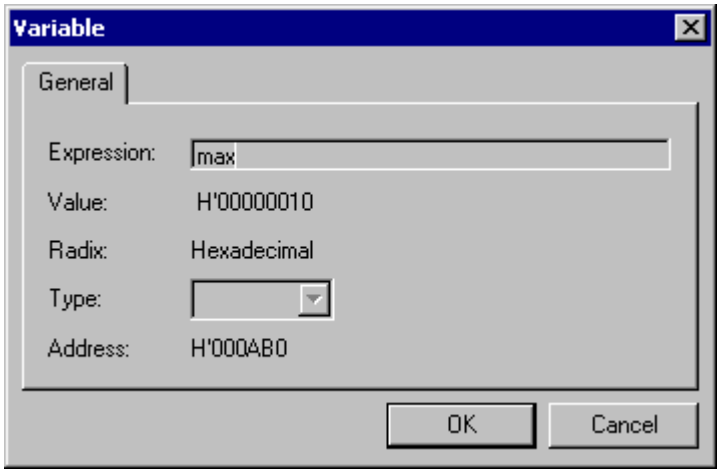
The symbol-address of the selected expression is displayed.

4.3.9.6 Property-Watch Window

This section explains the property in the Watch window.

■ Property: General

Figure 4.3-19 Property-Watch Window (General)



-Expression

The selected item is displayed.

-Value

The value of the selected expression is displayed.

-Base

The base in which the value is given is displayed.

-Type

The type of the selected expression is displayed.

-Address

The symbol-address of the selected expression is displayed.

4.4 View Menu

The View Menu displays each window. It also sets the tool bar and status bar to view or nonview.

■ Window view/non-view

The View Menu can set the following windows to view or non-view.

-Project Window

-Output Window

■ Window display (debug session)

The View Menu displays the following Debugger windows (valid only when SOFTUNE WORKBENCH is in the debug session):

-Symbol

-Assembly

-Register

-Memory

-Local

-Watch

-Trace

-Coverage

-Performance

-Command

-Object

-Realtime memory

-RAM Checker

■ Tool bar/status bar

The View Menu can set the tool bar and status bar to view or non-view. The set of tool buttons displayed in the tool bar can be also selected using the View Menu.

-Tool Bar

-Status Bar

■ Display/non-display the window switching using the tab

This function enables to choose whether the tab is displayed or non-displayed for switching. By using the switching tabs, it enables to switch edit, source, memory or watch windows easily.

-Tab

■ Font-related items

The fonts for each window can be changed.

-Fonts

4.4.1 Project/Output

"Project" switches the Project Window to view or non-view or vice versa. "Output" switches the Output Window to view or non-view or vice versa.

■ Project Window

When the Project Window is displayed, a check mark is set to the left of [Project] in the View Menu. Even if the Project window is switched to non-view, the project is not closed. When it is switched to display again, the Project Window is displayed at the previously displayed position.

■ Output Window

When the Output Window is viewed, a check mark is set to the left of [Output Window] in the View Menu. Even if the Output Window is switched to non-view, data in the Output Window is not cleared. When it is switched to displayed again, the Output Window is displayed at the previously displayed position.

If make, build, compile, assemble, update of dependencies or stop is executed when the Output Window is set to non-view, the Output Window automatically enters the view status.

Even if the Output Window is set to non-view when make, build, compile, or assemble is being executed, error messages are never lost. These messages are all displayed when the Output Window is redisplayed.

4.4.2 Symbol

"Symbol" views the Symbol Window.

■ Symbol

When the Symbol Window is opened, all the symbols used in the target file are displayed in the tree format.
This function (command) can only be used when SOFTUNE WORKBENCH is in the debug session.
Selecting [Symbol] when the Symbol Window has been already opened activates the Window.

4.4.3 Assembly

"Assembly" displays the Disassemble Window.

■ Assembly

When the Assembly window is opened, the assembly is displayed, starting at the specified address. Line assembly from the shortcut menu is also possible.

This function (command) can be used only when SOFTUNE WORKBENCH is in the debug session.

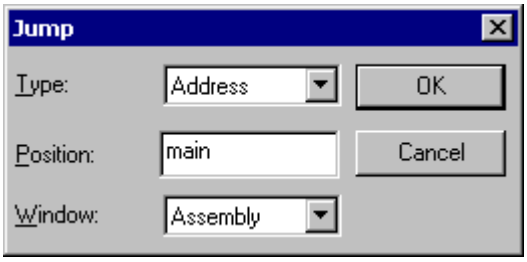
- When the Assembly Window has been already opened

  The Assembly Window is activated.

- When the Assembly Window is not opened

  The dialog for specifying the display start address (Figure 4.4-1) opens. Specify the address where display is to be started, then click the [OK] button.

Figure 4.4-1 Dialog Box for Specifying Display Start Address



- Type (Line number/Address/Frame)

  Specify the type of display position.

- Position

  Display disassembling from the position specified in the above type.

- Window

  To specify the window where jump is implemented.

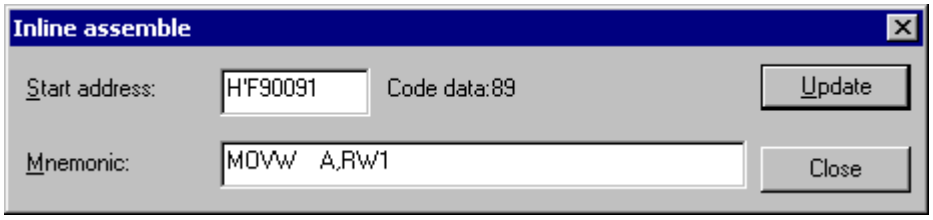
  For details, see Section "4.3.6 Jump".

  If it is a disassemble window, default will be shown as [Assembly].

■ Line assemble

Selecting [Line Assemble] from the shortcut menu opens the line assemble dialog shown in Figure 4.4-2.

Figure 4.4-2 Inline Assemble Dialog Box



Writing a mnemonic in the [Mnemonic] edit box and clicking the [Update] button assembles and sets the mnemonic, starting from the start address. The start address subsequently advances to the next address.

To change the address where the mnemonic is to be written, change [Start Address].

When mnemonic change is completed, click the [Close] button.

4.4.4 Register

"Register" displays the Register Window.

■ Register

When the Register Window is opened, the selected target MCU register name and each register retention value are displayed.

This function (command) can be used only when SOFTUNE WORKBENCH is in the debug session.

Selecting [Register] when the Register Window has been already opened activates the Window.

■ Changing register values

The values saved by the registers displayed in the Register Window can be changed directly by the following procedures:

- Full change

1. Double-click a register name or display value.
   - The register name or display value is reverse displayed.
2. Specify the values to set in turn using a hexadecimal number, starting from the highest-order digit (leftmost digit).
   -The register value is set automatically when the digit in the lowest-order bit (rightmost digit) is changed.

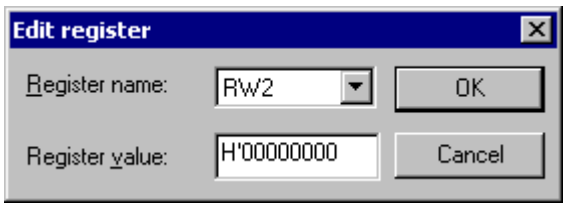
- Partial digital (bit field) change

1. Click the digit to be changed in the register value display.
   - The cursor appears at the left of the clicked digit.
2. Set the new value as a 1-digit hexadecimal number.
   - The cursor automatically moves to the right digit. If the changed digit is the lowest-order digit, the register value is updated automatically.
3. When changing of the required number of digits is completed, click another register name or register value display.
   - If the Register Window is closed without clicking another register name or register value display, the changed value is not set in the register.
   - This operation is not necessary when the change to the lowest-order bit is completed.

- Selected register name change

1. Click the right mouse button in the Register Window to display the shortcut menu, then select [Edit]. The register edit dialog shown in Figure 4.4-3 opens.
2. Select a register name.
3. Input the value to set.
4. Click the [OK] button.

Figure 4.4-3 Register Edit Dialog Box



- Register name

  Specify a register name.

- Register value

  Specify the value set in a register.

4.4.5 Memory

"Memory" displays the Memory Window.

■ Displaying memory data

When the Memory Window is opened, memory data dump is displayed, starting from the specified address. Memory data can be also modified using the shortcut menu.

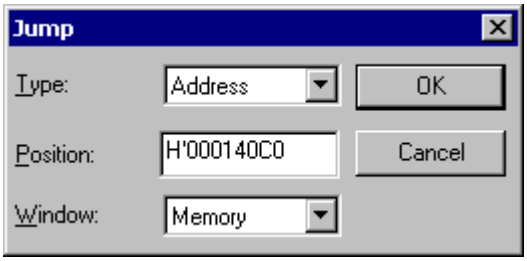
This function (command) can be used only when SOFTUNE WORKBENCH is in the debug session.

- When the Memory Window has been already opened

  The Memory Window is activated.

- When the Memory Window is not open, the dialog for specifying the display start address (Figure 4.4-4) opens.

Figure 4.4-4 Dialog for Specifying Display Start Address



- Type (Line number/Address/Frame)

  Specify the type of display position.

- Position

  Display disassembling from the position specified in the above type.

- Window

  To specify the window where jump is implemented.

  For details, see Section "4.3.6 Jump".

  If it is a disassemble window, default will be shown as [Assembly].

■ Changing memory data

Memory data can be changed by rewriting the displayed dump value directly. When a character string is entered from the ASCII character string display field, the ASCII code of each character is set automatically in the corresponding address.

The address where the memory dump is to be started can be changed by changing the address field display.

■ Add bookmark

Please click on the right button on memory window to display short cut menu. Then, select [Add bookmark]. [Add bookmark] Dialog will be displayed. (Figure 4.4-5, Figure 4.4-6)

Figure 4.4-5 Add Bookmark Dialog (Address)

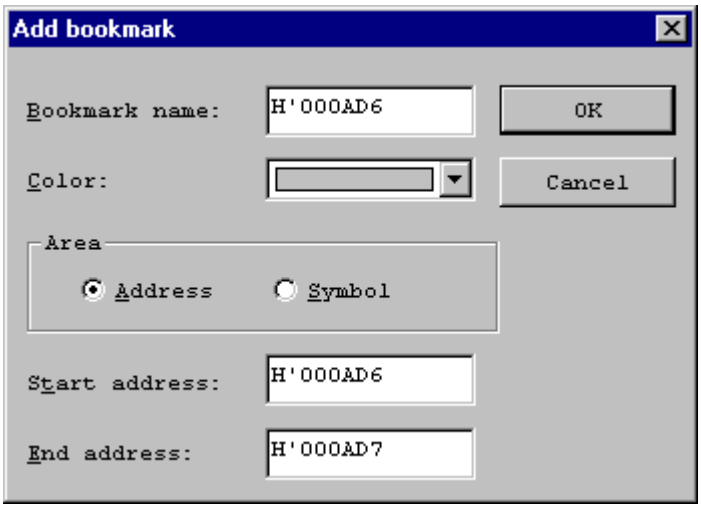
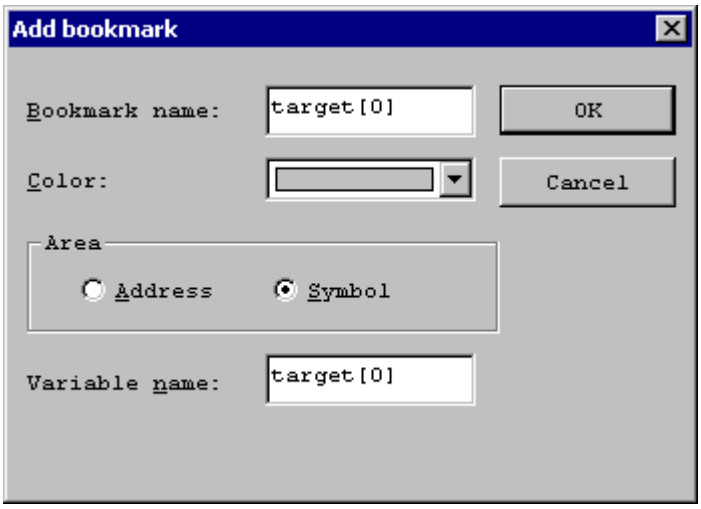


Figure 4.4-6 Add Bookmark Dialog (Symbol)



- Bookmark name

  To specify the name of bookmark. Default will be changed as follows depending on setup made in [Area].

  If the [Area] is [Address]: Start address

  If the [Area] is [Symbol]: Variable name

- Color

  To specify the color of background in bookmark.

- Area

  To specify the method for determining the address range of bookmark.

  Address: Specify with start address or end address

Symbol: Specify the address range with the name of symbol

- Start address

  To specify the start address of bookmark.

- End address

  To specify the end address of bookmark.

- Variable name

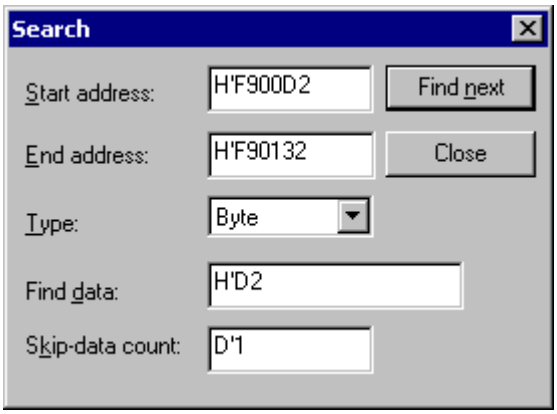
  To specify the name of symbol for setting a bookmark.

■ Searching memory data

Click the right mouse button in the Memory Window to display the shortcut menu, then select [Search] from the menu.

The search dialog shown in Figure 4.4-7 opens.

Figure 4.4-7 Search Dialog



-Start Address

Specifies start address of search range

-End Address

Specifies end address of search range

-Type

Selects data type (Byte/Word/Long/Ascii)

-Find Data

Specifies matching data type.

When ASCII is selected as the data type, specify a character string. To search for several data items continuously when the data type is not ASCII, write each item, delimited by a comma (,).

-Skip Data Count

To search the search range continuously, set 1. When a value greater than or equal to 2 is set as the skip byte count, addresses are skipped for each set count and the search range is searched.

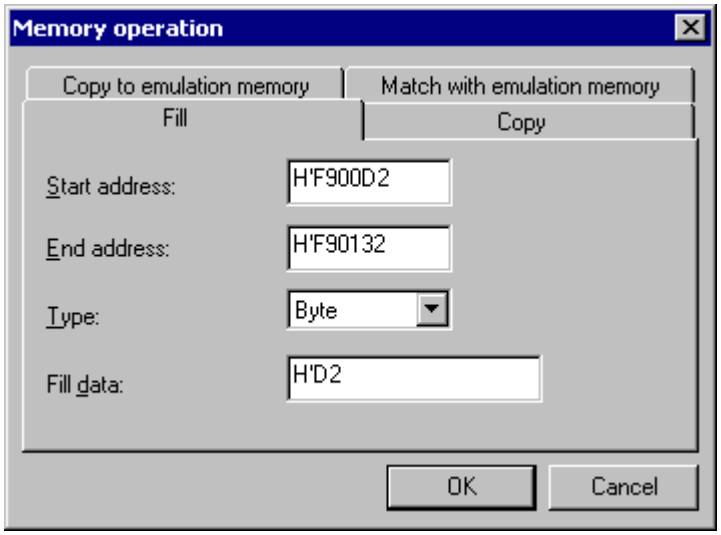
■ Special operation

To fill memory with data or to copy data to memory, start debugging, click the right mouse button in the Memory Window to display the menu, then select [Special Operation] from the menu. The memory operation dialog opens.

Clicking the [OK] button from this dialog starts the function of the open tab.

-Fill

Figure 4.4-8 Memory operation Dialog (Fill)



To fill memory with data, open the [Fill] tag, then set the start address, end address, data type, and filling data.

-Start Address

Specifies start address of memory area to be filled with data. Data filling is started at this address.

-End Address

Specifies end address of memory area to be filled with data. Data filling is continued to this address.

-Type

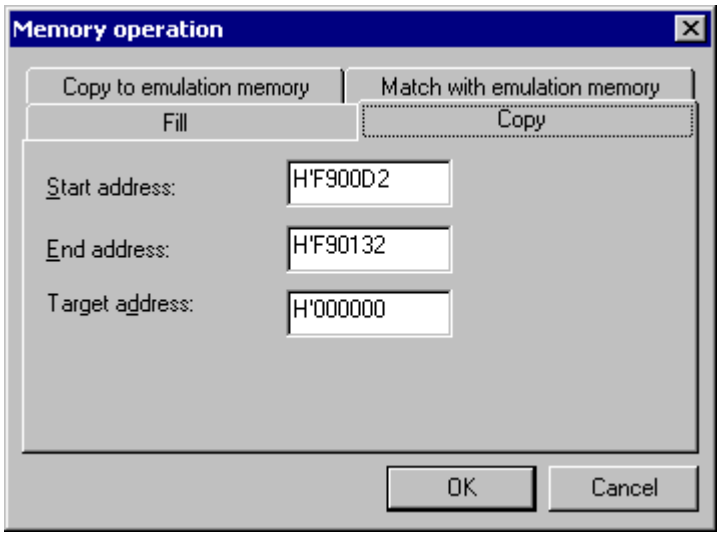
Specifies type of filled data. (Byte/Word/Long/Ascii)

-Fill Data

Specifies filling data to fill specified memory area. Several filling data can be specified, delimited by a comma (,).

-Copy

Figure 4.4-9 Memory Operation (Copy)



To copy data to memory, open the [Copy] tag, then set the copy source start address, copy source end address, and copy destination start address.

-Start Address

Specifies start address of copy source area. Data copy is started at this address.

-End Address

Specifies end address of copy source area. Data copy is continued to this address.

-Target Address

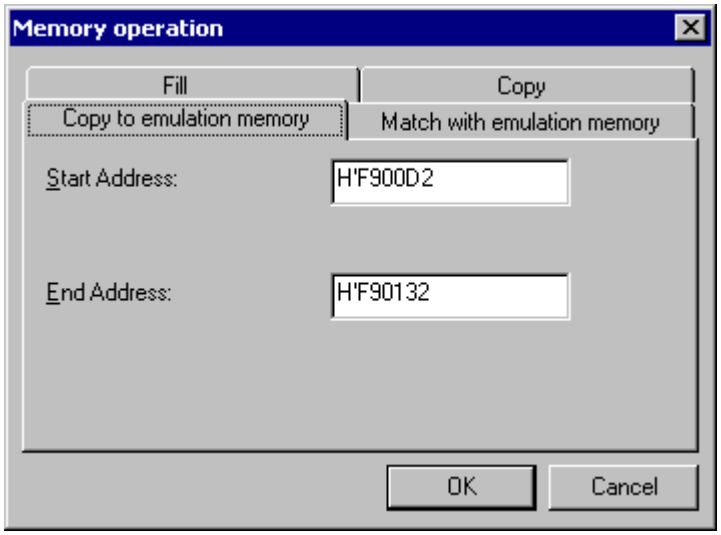
Specifies start address of copy destination area.

Note:

Copy to emulation memory is valid only at emulation debugging.

-Copy to emulation memory

Figure 4.4-10 Memory Operation Dialog (Copy Emulation Memory)



To copy data from user memory to emulation memory, open the [Copy to Emulation Memory] tag, then set the transfer start address and transfer end address.

-Start Address

Specifies address at which data transfer is to be started. Copy to emulation memory is started at this address.

-End Address

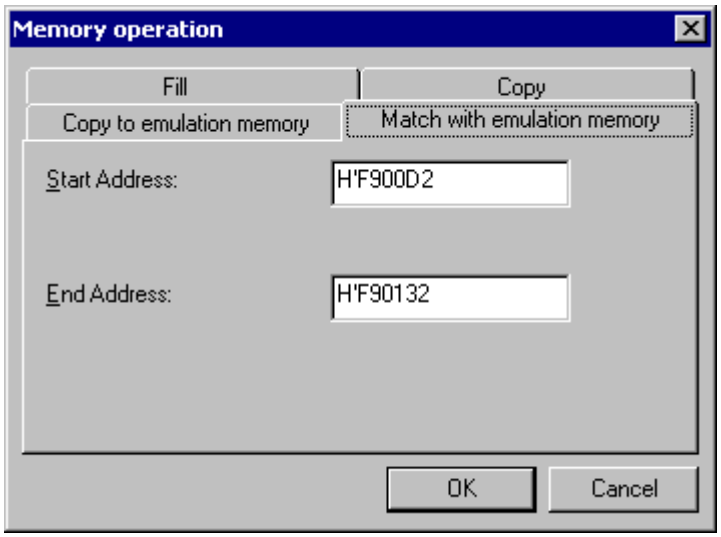
Specifies end address of transfer source area. Copy to emulation memory is continued to this address.

Note:

Collating user memory with emulation memory is valid only at emulation debugging.

-Match with emulation memory

Figure 4.4-11 Memory Operation Dialog (Match with Emulation Memory)



To collate user memory with emulation memory, open the [Match with Emulation Memory] tag, then set the collocation start address and collation end address.

-Start Address

Specify the address at which collation is to be started. Collating user memory with emulation memory is started at this address.

-End Address

Specify the end address of the collation source area. Collating user memory with emulation memory is continued to this address.

■ Comparing memory blocks

To compare memory blocks, click the right mouse button in the Memory Window to display the menu, then select [Compare] from the menu. The comparison dialog shown in Figure 4.4-12 opens.

-Start Address

Specifies start address of comparison source area. Memory block comparison is started from this address.

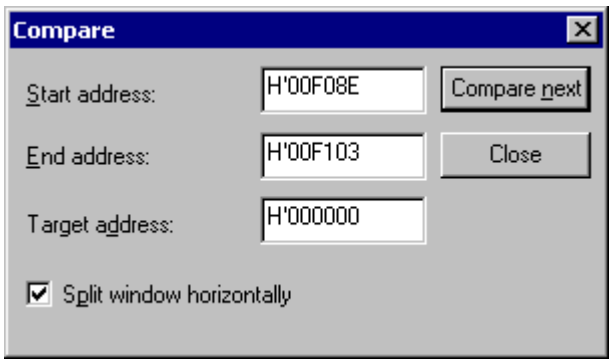
-End Address

Specifies end address of comparison source area. Memory block comparison is continued up to this address.

-Target Address

Specifies start address of comparison destination area.

Figure 4.4-12 Comparison Dialog



For example, when 4 is specified as the skip byte count, addresses are skipped for each 4 bytes like address 4 and address 8 and the search range is searched.

[Example]

Suppose the data in memory is "00000000 01 02 03 01 02 01 02 03 04 01 01 02 03".

When Data Type is Byte, Search Data is 01, and Skip Byte Count is 1, all search data is found. However, when Skip Byte Count is 2, only 01 search data at address 00000000 and 0000000A is found. 01 search data at other addresses is not found.

When Data Type is Byte, Search Data is 01 and 02, and Skip

Byte Count is 3, only 01 and 02 search data at address 00000000 and address 00000003 is found. 01 and 02 search data at address 00000005 and address 0000000A is not found.

■ Edit

To edit memory data, click the right mouse button in the Memory Window to display the menu, then select [Edit] from the menu. The edit dialog shown in Figure 4.4-13 opens.

-Address

Specifies address to edit.

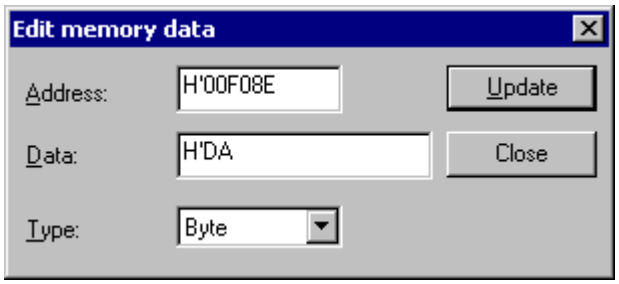
-Data

Specifies memory data to rewrite.

-Type

Specifies size of data to rewrite. (Byte/Halfword/Word/Double Word)

Figure 4.4-13 Memory Edit Dialog



■ Display setup

To set the display format of the Memory Window, click the right mouse button in the Memory Window, then select [Display Setup] from the menu. The display setup dialog shown in Figure 4.4-14 opens.

-Display format

Specifies display format of Memory Window.

"Bit", "byte", "word", or "long" can be selected.

-Display Format

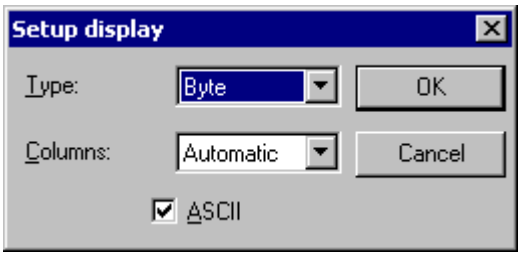
Selects whether to display ASCII characters at right of Memory Window.

-Columns

To specify the number of bytes display in a line.

The number of bytes can be chosen from Automatic, 4byte, 8byte, 16byte, 32byte or 64byte.

Figure 4.4-14 Display Setup Dialog Box



4.4.6 Local

"Local" displays the Local Variable Window.

■ Local Variable Window

The Local Variable Window displays, in tree format, the local variables of the function where the current instruction pointer exists using the function name as the root. The displayed variables cannot be added nor can the displayed variables be canceled.

The variable values modified as a result of program execution are updated automatically. For this reason, the user can observe how variable values change as a result of program execution.

Variable values can be also modified to continue debugging.

■ Setting a base number

The base number when a variable value is displayed can be set for each variable using the following procedure:

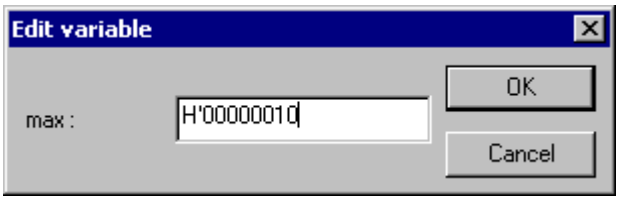
1. Click the right mouse button in the line containing the variable of the base number to be modified.
   The shortcut menu is displayed.
2. Move the mouse cursor to [Base Number].
   The list showing selectable base numbers is displayed in the submenu.
3. Select the base number to modify from the list.

■ Modifying a variable value

The variable value can be modified by the following procedure:

1. Click the right mouse button in the line containing the variable to be modified.
   The shortcut menu is displayed.
2. Select [Edit] from the shortcut menu.
   The variable edit dialog shown in Figure 4.4-15 opens.
3. Set a variable value, then click the [OK] button.

Figure 4.4-15 Variable Edit Dialog



4.4.7 Watch

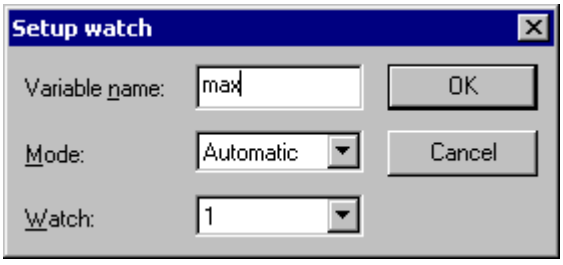
"Watch" displays the Watch Window.

■ Watch Window

The Watch Window displays the values of the specified variables in tree format. The variable values modified as a result of program execution are updated automatically. For this reason, the user can observe how variable values change as a result of program execution. The shortcut menu can be used to modify the displayed variable values.

1. Click the right mouse button in the Watch Window.
   The shortcut menu is displayed.
2. Click [Setup].
   The watch setup dialog shown in Figure 4.4-16 opens.
3. Input a variable name from the dialog. Also select a [mode] as required.
4. Click the [OK] button.

Figure 4.4-16 Watch Setup Dialog



- Variable name

  Specify the name of a variable to be displayed.

- Mode (Automatic/ C language/Assembler)

  Specify the mode, C or assembler language, in which a variable is displayed. For automatic operation, the variable is displayed in the predetermined language mode.

- Watch

  Specify the Watch Window Number name.

■ Setting a base number

The base number when a variable value is displayed can be set for each variable using the following procedure:

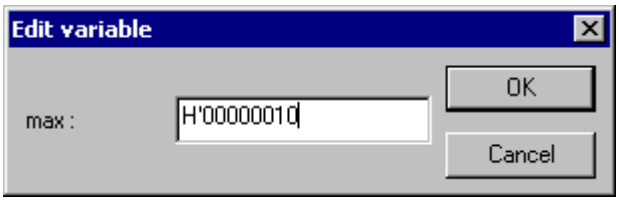
1. Click the right mouse button in the line containing the variable of the base number to modify.
   The shortcut menu is displayed.
2. Move the mouse cursor to [Base Number].
   The list showing selectable base numbers is displayed in the submenu.
3. Select the base number to modify from the list.

■ Modifying a variable value

The variable value can be modified using the following procedure:

1. Click the right mouse button in the line containing the variable to modify.
   The shortcut menu is displayed.
2. Select [Edit] from the shortcut menu.
   The variable edit dialog shown in Figure 4.4-17 opens.
3. Set a variable value, then click the [OK] button.

Figure 4.4-17 Variable Edit Dialog



Canceling display of unnecessary variable

Display of a set variable that is no longer needed can be canceled from the Watch Window using the following procedure:

1. Click the right mouse button in the line displaying the variable to cancel.
   The shortcut menu is displayed.
2. Select [Delete] from the shortcut menu.

Note:

Display of each array element or each member (e.g., structure) cannot be canceled. When [delete] is executed in each element or member line, display of the array or structure is canceled.

4.4.8 Trace

"Trace" displays the Trace Window.

■ Trace

This function retroactively displays addresses and instructions executed so far. [Enable] can be switched to [Disable] or vice versa during debugging. Display per machine instruction, cycle display, display per source can be selected as trace result display.

■ Update

The Trace Window display is not updated in real-time according as debugging progresses. Consequently, to display the latest trace, click the right mouse button in the Trace Window to display the shortcut menu, then select [Update] from the menu.

■ Buffer size

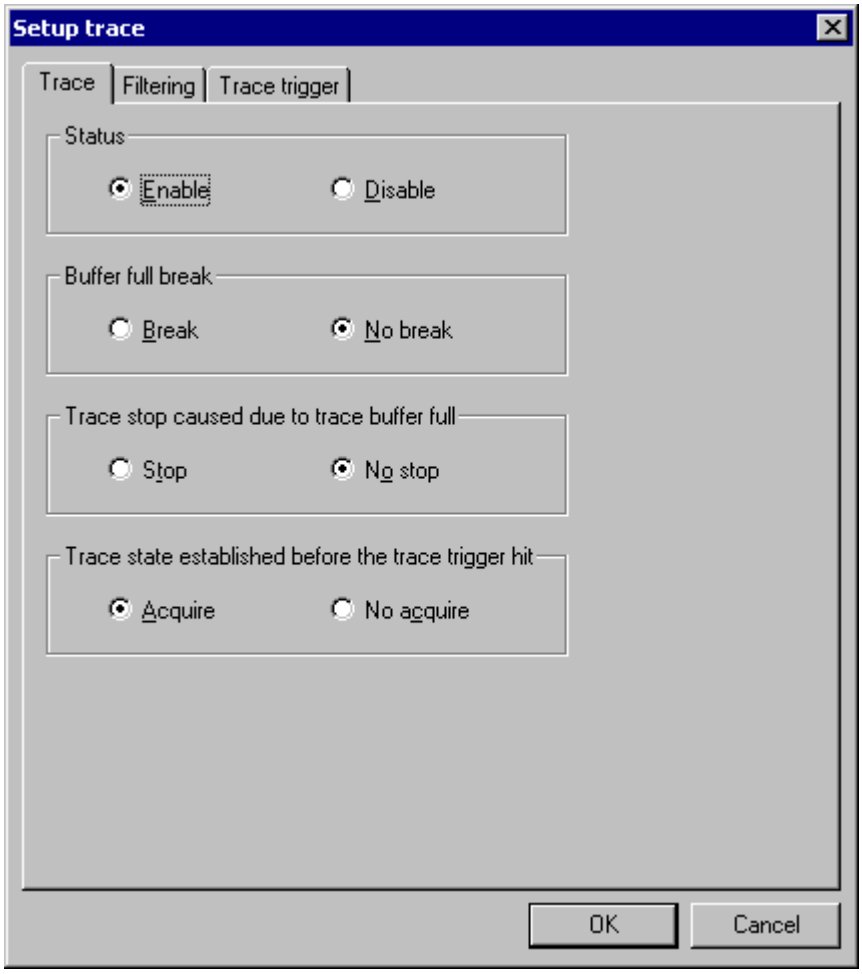
Trace data is buffered in the trace buffer. The trace buffer becomes full some time during debugging because its size is finite. When the trace buffer becomes full, the program being executed can be stopped.

■ Trace setup

Select [Setup] from the shortcut menu.

-Trace

Figure 4.4-18 Trace Setup Dialog



-Status

Specifies control status.

-Buffer Full Break

Specifies whether or not to break trace buffer full.

-Trace stop caused due to trace buffer full

Specifies whether or not to halt trace due to trace buffer full.

This item is only displayed for MB2147-01 and MB2147-05.

-Trace state established before the trace trigger hit

Specifies whether or not to acquire the trace state before a trace trigger hit occurs.

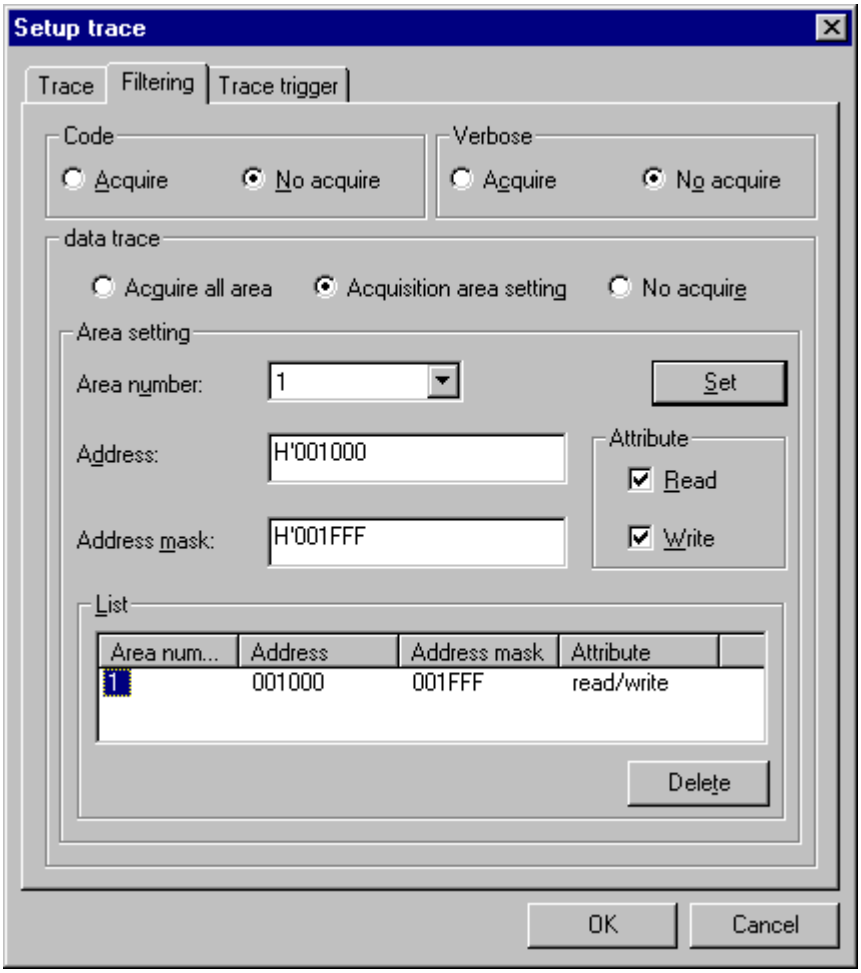
This item is only displayed for MB2147-01 in single trace mode.

Note:

In single trace and multitrace, items are individually set.

-Filtering [MB2147-01, MB2147-05]

Figure 4.4-19 Filtering (MB2147-01,MB2147-05)



- Code

  Whether or not to acquire the instruction trace is specified.

- Verbose

  Whether or not to acquire the redundant data is specified.

- Status

  The trace data acquisition state is specified.

  Acquire all area : Trace data with read/write attribute is acquired in the entire area.

  Acquisition area setting : The acquisition area for trace data with read/write attribute is set to acquire the trace data.

  Not acquire : Trace data with read/write attribute is not acquired in the entire area.

- Area number

  The area number (1 or 2) to set the range of filtering is specified.

- Address

  The address for which filtering is to be performed is specified.

- Address mask

  The mask value for the address for which filtering is to be performed is specified.

- Attribute

  The attribute of the data for which filtering is to be performed is specified.

- List

  The setting (the set data) of the filtering area is displayed.

- [Set] button

  This button sets the filtering area based on the specified address.

- [Delete] button

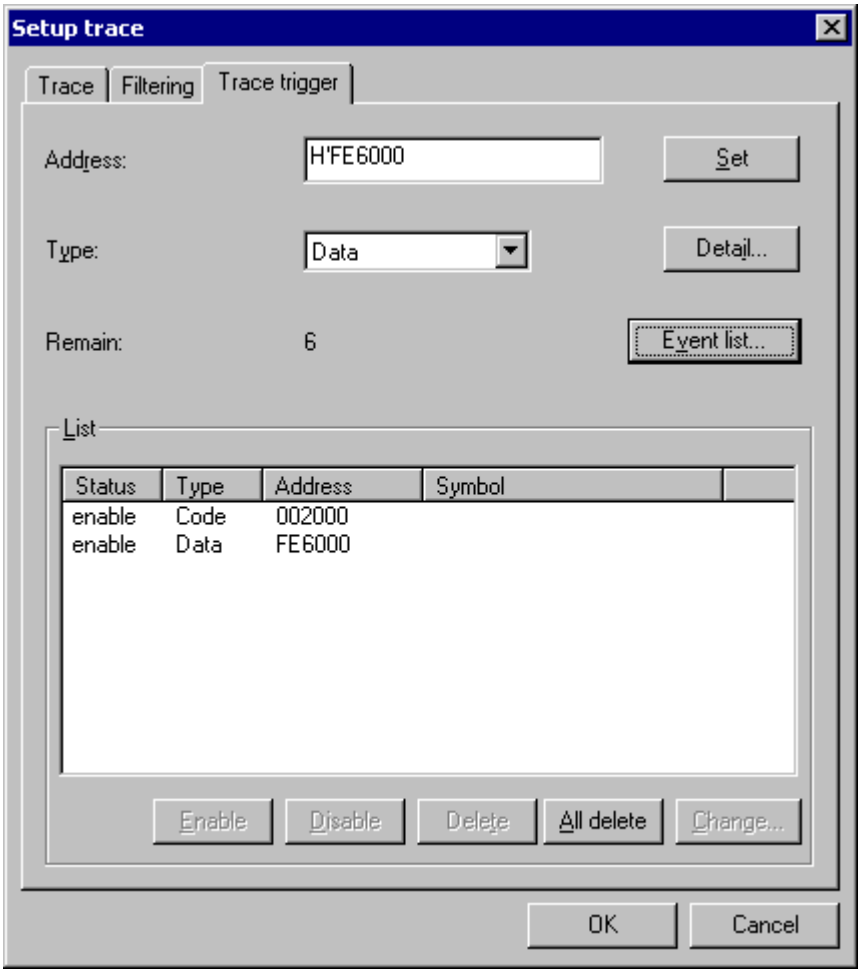
  This button deletes the specified filtering area.

Note:

If the debug function for MB2147-01 is in the Trace Enhancement mode, filtering can be configured even during the execution of user program except during the process of trace execution. For more information, refer to "Trace Control When a User Program is Running" described later.

- Trace trigger [MB2147-01]

Figure 4.4-20 Trace Trigger (MB2147-01)



- Address

  In this field, the address or symbol that sets a trigger point is specified.

- Type

  In this field, the type (code/data) of the trace trigger is specified.

- Remain

  In this field, the remaining count of settable trace trigger types currently being selected is displayed.

- List

  The data of the trace trigger currently being set are displayed.

- [Set] button

  This button is used to set the trace trigger at the set address.

- [Details] button

  This button is used to display the trace trigger details setting dialog used to set the detailed condition other than addresses.

- [Event List] button

  This button is used to display the event list dialog used to check the setting of all events.

- [Enable] button

  This button is used to enable the trace triggers in the trace trigger list currently being selected.

- [Disable] button

  This button is used to disable the trace triggers in the trace trigger list currently being selected. The trace triggers are simply disabled; that is, the setting itself of the trace triggers is not cancelled.

- [Delete] button

  This button is used to delete the setting of the trace triggers in the trace trigger list that are currently selected.

- [All delete] button

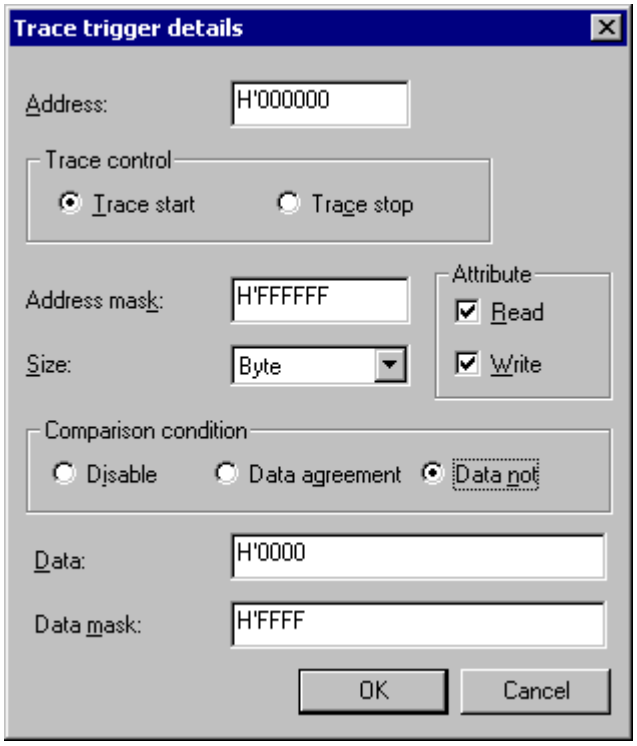
  This button is used to delete the setting of all the trace triggers in the trace trigger list.

- [Change] button

  This button is used to change the setting of the trace triggers in the trace trigger list currently being selected.

- Trace trigger details setting dialog [MB2147-01]

Figure 4.4-21 Trace Trigger Details Setting Dialog



- Address

  The address that sets a trace trigger is specified.

- Trace control

  The trace operation to be performed when the set trace trigger is hit is specified.

- Address mark

  The mask value for the specified address is specified.

- Size (byte/word)

  The data size during data access is specified.

- Attribute

  The attribute during data access is specified.

- Comparison condition

  The data comparison condition is specified.

  Disable : No data is specified for the trigger condition.

  Data match : Data match (data matches the specified data) is specified for the trigger condition.

  Data NOT : Data mismatch (data does not match the specified data) is specified for the trigger condition.

- Data

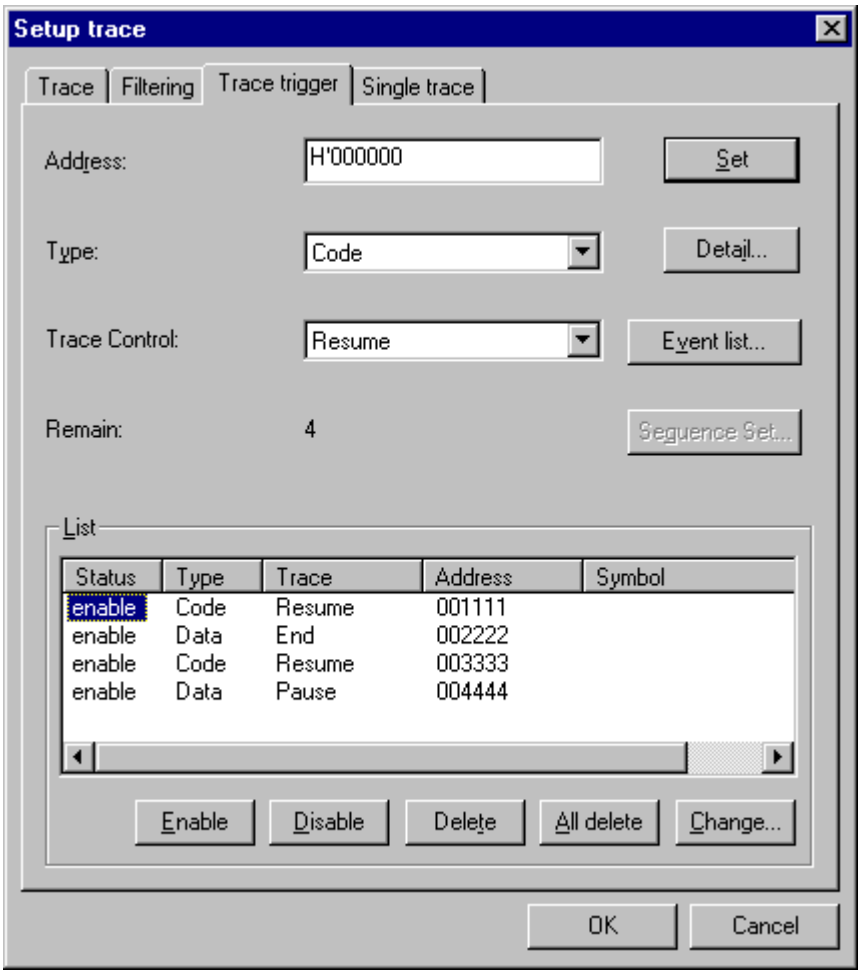
  Data to be used for the trigger condition is set.

- Data mask

  The mask value for the specified data is specified.

- Trace trigger [MB2147-01]

Figure 4.4-22 Trace Trigger (MB2147-01)



- Type

  Specifies the type of the trace trigger.
  (Code/ Data/ Data watch/ Sequencer)

- Trace Control

  Specifies the operation of the trace acquisition when it hits the specified address.

  Resume: Resumes the trace acquisition (from a pause state).

  Pause: Pauses the trace acquisition.

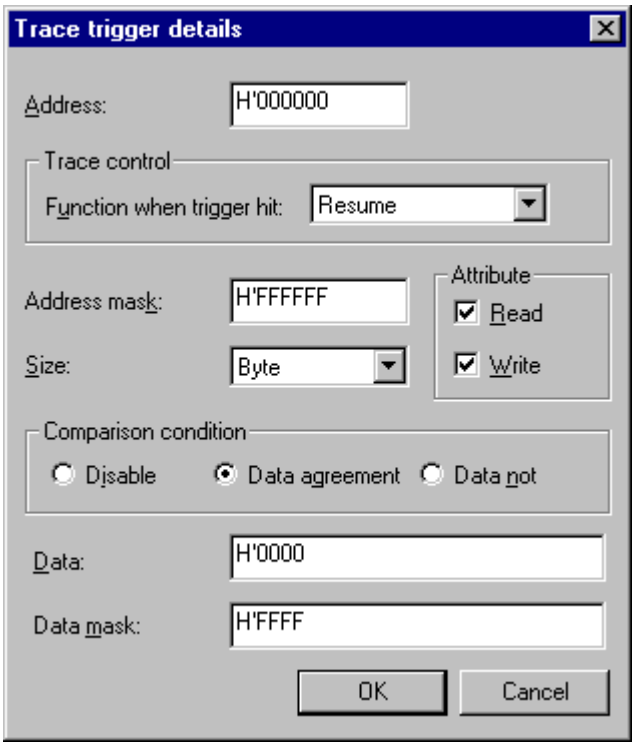
  End: Ends the tracing.

Notes:

1. If the debug function for MB2147-01 is in the Trace Enhancement mode, trace trigger can be configured even during the execution of user program except during the process of trace execution. For more information, refer to "Trace Control When a User Program is Running" described later.
2. "Tracing" means that the trace acquisition is either in "Tracing" or "Pause" state.

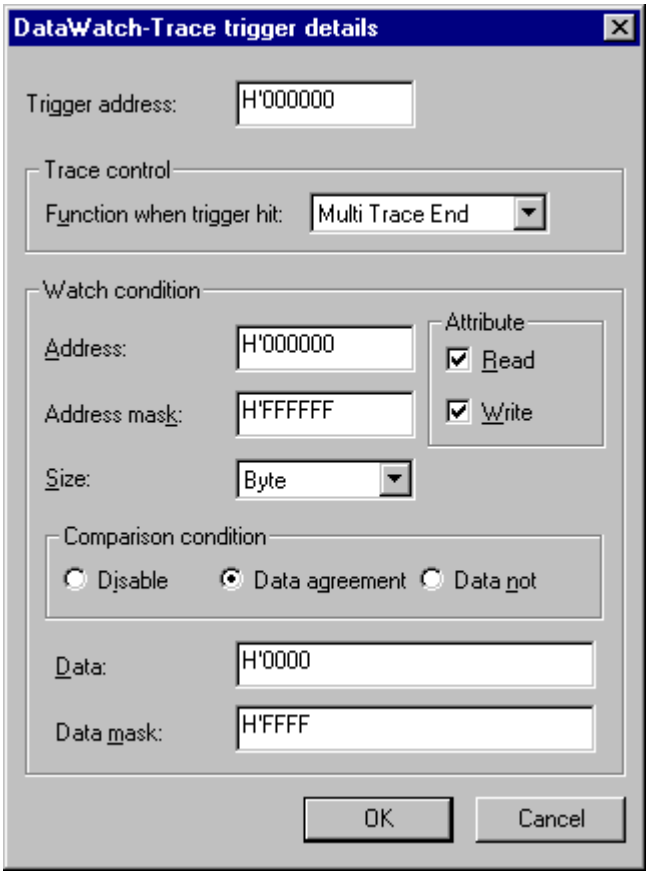
- Trace Trigger Details Setting Dialog (code/data) [MB2147-01]

Figure 4.4-23 Trace Trigger Details Setting Dialog (MB2147-01)



- DataWatch - Trace Trigger Details [MB2147-01]

Figure 4.4-24 DataWatch - Trace Trigger Details Dialog (MB2147-01)



- Trigger address

  This field specifies the address to which set a break point.

Trace control

- Function when trigger hit

  Specifies the operation of the trace acquisition when it hits the specified address.

  <In single trace mode>

  Resume: Resumes the trace acquisition (from a pause state).

  Pause: Pauses the trace acquisition.

  End: Ends the tracing.

  <In multi trace mode>

  Trace End: Transits the trace buffer to next block.

  Multi Trace End: Ends the trace acquisition.

Watch condition

- Address

  This field specifies the address or symbol to be watched.

- Address mask

  This field specifies the mask value that corresponds to the specified address.
- Size (Byte/Word)

  Specifies the data size when data accessing.
- Attribute

  Specifies the attributes when data accessing.
- Comparison condition

  Specifies the data comparison conditions.

  Disable: Doesn't specify any data for the trigger condition.

  Data agreement: Trigger condition is when the address matches with the specified data.

  Data not: Trigger condition is when the address doesn't match with the specified data.
- Data

  This field sets the data when data accessing.
- Data mask

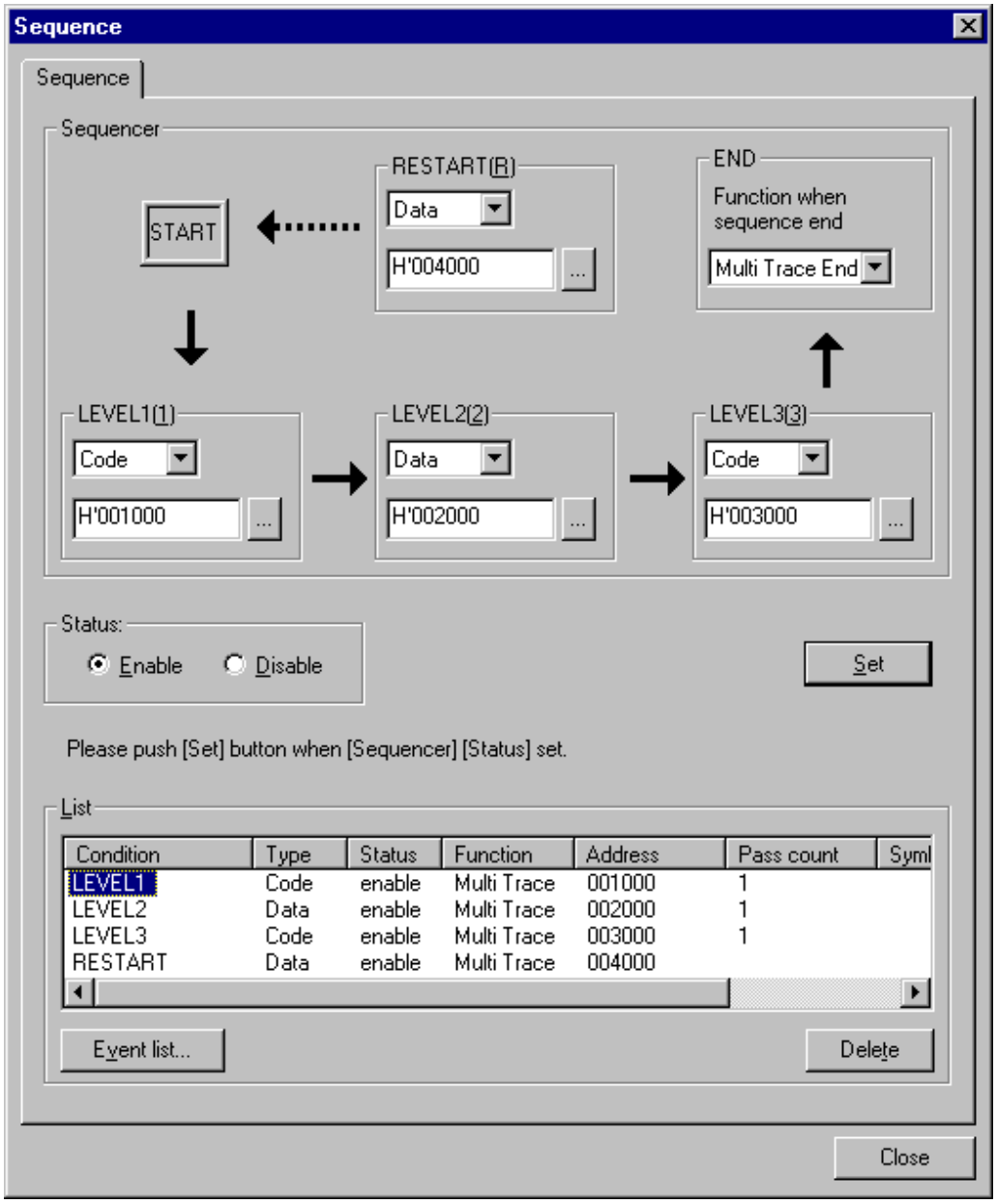
  This field specifies the mask value that corresponds to the specified data.

Notes:

1. Data Watch Trace trigger becomes valid only when the debug function of MB2147-01 is in Trace Enhancement mode. For more information, please refer to Section "2.3.1.6 Debug Function" in SOFTUNE Workbench User's Manual.
2. You can set the Data Watch Trace trigger while running a user program, but tracing. For more information, refer to "■Trace Control When a User Program is Running" described later.
3. "Tracing" means that the trace acquisition is either in "Tracing" or "Pause" state.

- Trace Trigger Setting (Sequence) [MB2147-01]

Figure 4.4-25 Trace Trigger Setting Dialog (Sequence) (MB2147-01)



- Function when sequence end

  Specifies the operations of the sequencer when it reaches to the end.

  <In single trace mode>

  Break: Stop the execution of the program.

  Trace End: Ends the tracing.

  <In multi trace mode>

  Break: Stop the execution of the program.

  Trace End: Transits the trace buffer to next block.

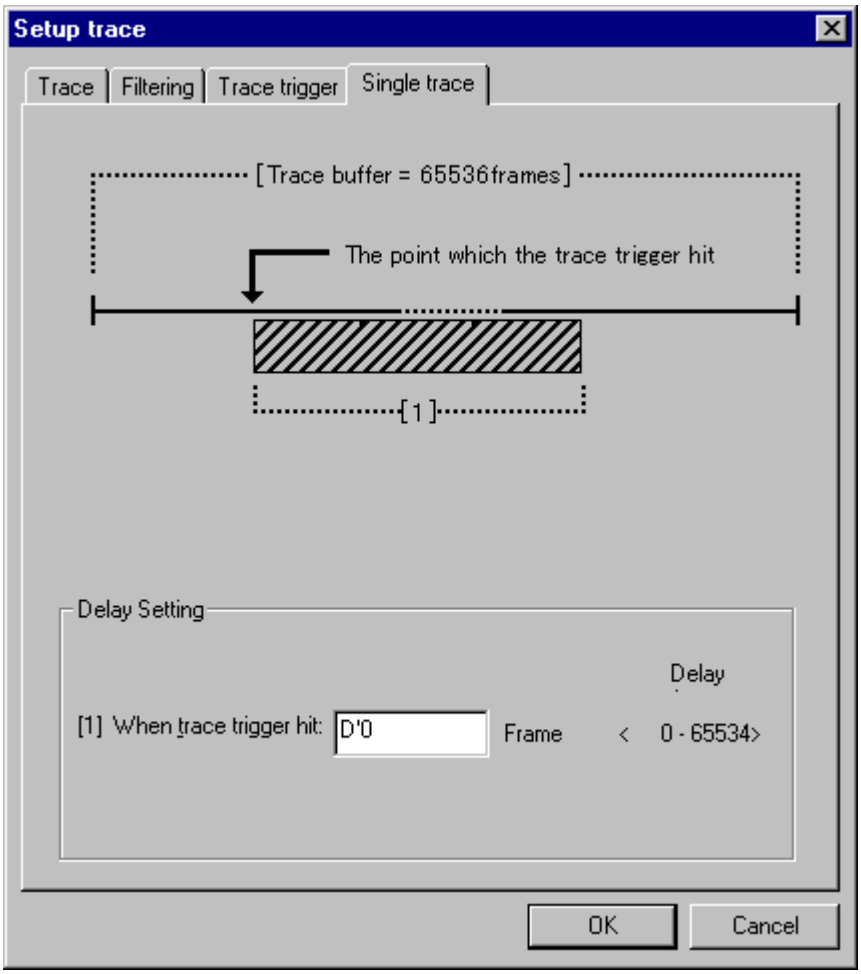
  Multi Trace End: Ends the trace acquisition.

Notes:

1. Sequence Trace trigger becomes valid only when the debug function of MB2147-01 is in Trace Enhancement mode. For more information, please refer to Section "2.3.1.6 Debug Function" in SOFTUNE Workbench User's Manual.
2. You can set the Sequence Trace trigger while running a user program, but tracing. For more information, refer to "■Trace Control When a User Program is Running" described later.
3. "Tracing" means that the trace acquisition is either in "Tracing" or "Pause" state.

- Single Trace [MB2147-01]

Figure 4.4-26 Single Trace Dialog (MB2147-01)



- Delay Setting
  - When trace trigger hit

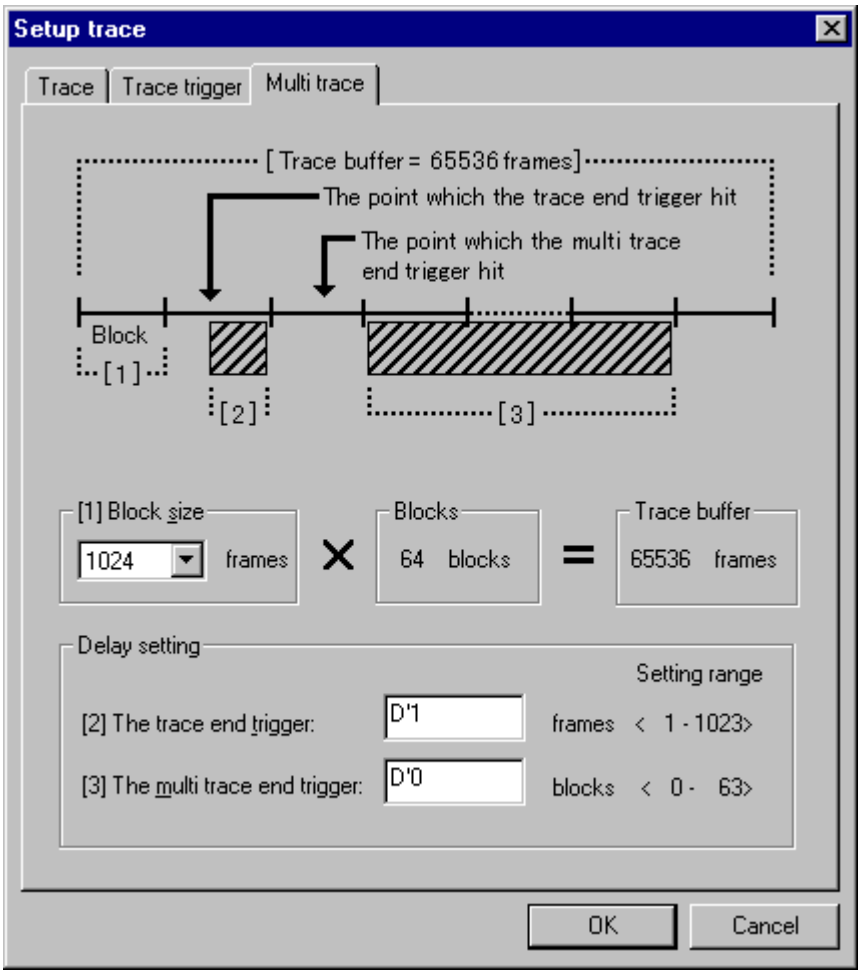
    Specifies the trace delay acquired from the point when the trace end trigger hits (in frames). You can set up to 65534 frames.

Note:

Single trace tab becomes valid only when the debug function of MB2147-01 is in Trace Enhancement mode. For more information, please refer to Section "2.3.1.6 Debug Function" in SOFTUNE Workbench User's Manual.

- Multi Trace [MB2147-01]

Figure 4.4-27 Multi Trace (MB2147-01)



- Block size

  Specifies the block size of the multi trace. You can choose from 128/ 256/ 512/ 1024 frame.

- Delay Setting
  - The trace end trigger

    Specifies the trace delay acquired from the point when the trace end trigger hits (in frames). Setting range may vary depending on the block size.

  - The multi trace end trigger

    Specifies the trace delay acquired from the point when the multi trace end trigger hits (in blocks). Setting range may vary depending on the number of blocks.

Note:

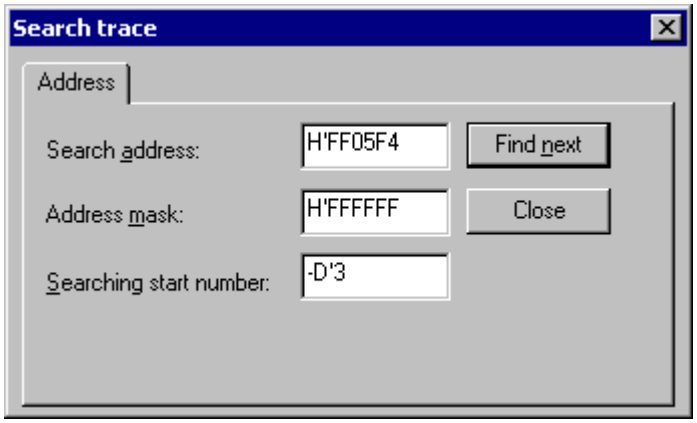
Multi trace tab becomes valid only when the debug function of MB2147-01 is in Trace Enhancement mode. For more information, please refer to Section "2.3.1.6 Debug Function" in SOFTUNE Workbench User's Manual.

■ Trace Search

Select [Search] from the shortcut menu.

-Address

Figure 4.4-28 Trace Search Dialog (Address)



-Search address

Specifies start address of search range.

-Address mask

Specifies mask address of search range.

-Searching Start Number

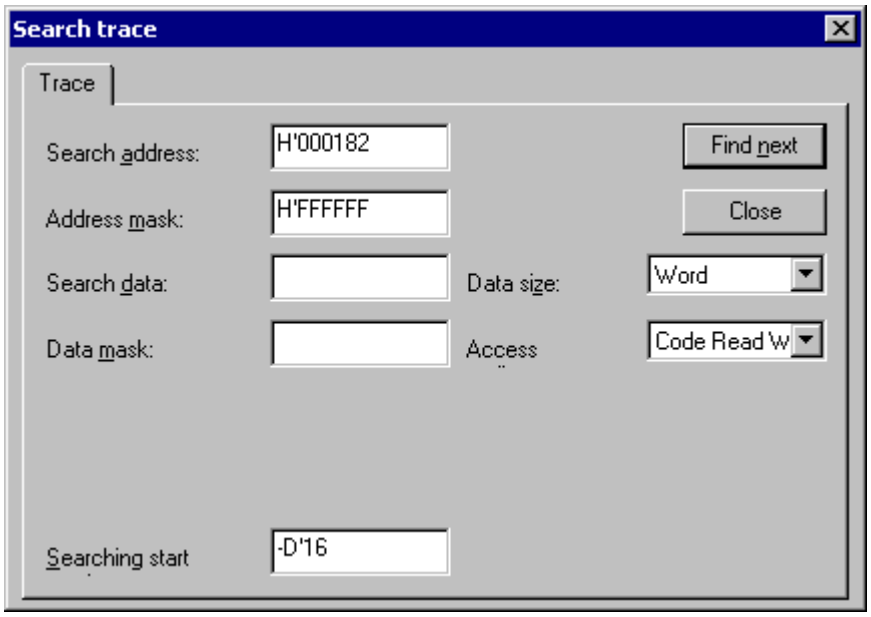
Specifies number of frame where search to start.

-Trace

Note:

Trace is valid only at emulation debugging.

Figure 4.4-29 Trace Search Dialog (EML)



-Search Address

Specifies address to search.

-Address Mask

Specifies address mask. Only bit-1 addresses are compared.

-Search Data

Specifies data to find.

-Data Mask

Compares only 1-bit data.

-Search Start Number

Specifies number of frame where search to start.

-Size

Specifies data access size (byte/word).

-Access Attribute

Specifies access attribute. (Code/Read/Write/Code Read/Read Write/Code Read Write)

-Level Change Point Search

Specifies sequencer level change point. This item is not displayed when multitrace search is executed.

■ Example of address mask

The relationship among search address, address mask and the actual address to be found is shown in Table 4.4-1.

Table 4.4-1 Relationship among Search Address, Address Mask, and the Actual Address to be found

Search address	Address mask	Actual address to be found
H'F0F0CA	H'FFFFFF	H'F0F0CA
	H'FFF000	Any address from H'000000 to H'F0FFFF

■ Back trace

Select [Back Trace] from the shortcut menu.

Figure 4.4-30 Back Trace Dialog

- Frame Number

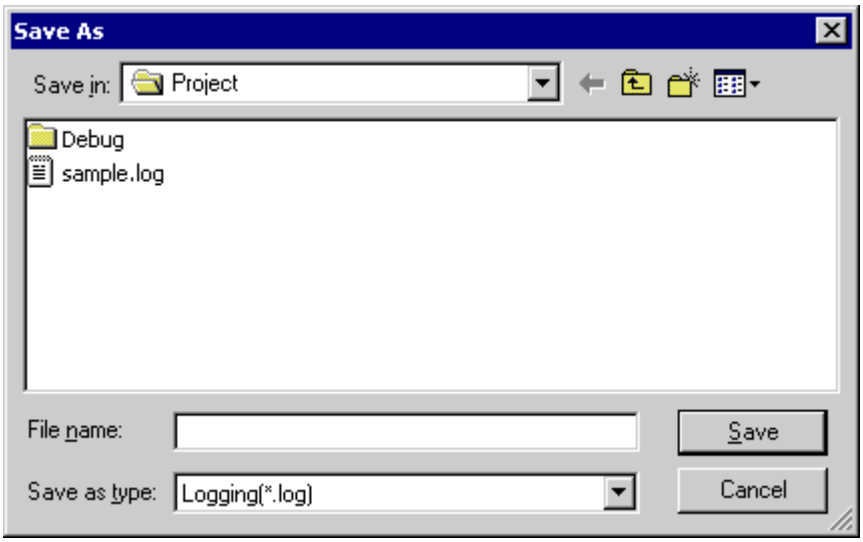
Specifies frame number corresponding to Trace Window.

■ Save

Select [Save] from the shortcut menu.

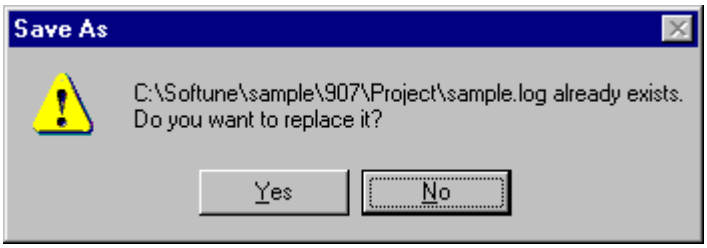
Save the trace data to the specified file.

Figure 4.4-31 Save Trace Data Dialog



When specifying a file name, specify an existing file, and select whether to save or not.

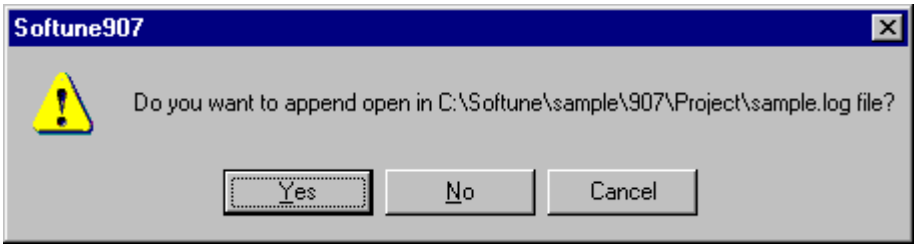
Figure 4.4-32 Trace Data Save Dialog



Select whether to add and save when selecting "Yes" (Save).

Selecting "No" does not save, trace data to the file.

Figure 4.4-33 Trace Data Save Dialog



Selecting "Yes" (Add/Save) adds and saves trace data to the file.

Selecting "No" saves trace data to in the file.

■ Switching of Event Modes

Select [Event mode] from the shortcut menu.

A dialog to switch the trace function between [Single trace] and [Multi trace] will appear.

Alternatively, this dialog can be displayed by selecting [Debug] ? [Set debug environment] ? [Debug environment] menu.

Figure 4.4-34 Event Mode Dialog

■ Trace Control When a User Program is Running

The following trace controls are available while running a user program:

1. Forced stop of tracing

   Forces to stop tracing temporarily when a trace is in execution.

   After the forced stop, you can resume tracing by a forced start.

   <Operating instructions>

   - Trace tool bar (Abort button)
   - [Abort] shortcut menu in the trace window

2. Forced start of tracing

   Resumes the tracing when it has been forced to stop.

   This is only operational while a tracing has been forced to stop.

   <Operating instructions>

   - Trace tool bar (Start button)
   - [Start] shortcut menu in the trace window

3. Setting of the Filtering

   You can set the filtering while running a user program, but tracing.

   If you set the filtering while tracing, you must force the tracing to stop. You are also required to execute "forced start" for the setting to take effect once the setting is done.

   <Operating instructions>

   - [Filtering] tab in the Setup trace dialog
   - SET DATATRACEAREA command

     Note: This is only available while not tracing.

4. Setting of the trace trigger

   (Code/ Data/ Data watch/ Sequencer)

   You can set the filtering while running a user program, but tracing.

   If you set the filtering while tracing, you must force the tracing to stop. You are also required to execute "forced start" for the setting to take effect once the setting is done.

   <Operating instructions>

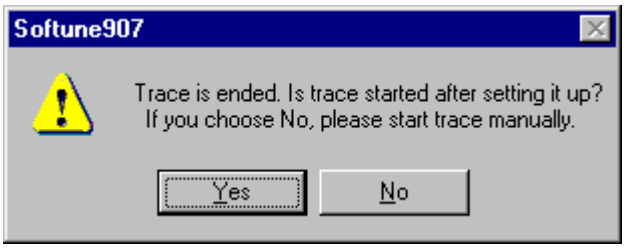
   - [Trace trigger] tab in the Setup trace dialog
   - SET TRACETRIGGER command

     Note: This is only available while not acquiring the trace.

If you attempt to set either Filtering or Trace trigger when a tracing and user program are in execution, the following dialogs appear.

● When tracing has been stopped

Figure 4.4-35 The setup confirmation dialog when running a user program (During the trace acquisition is stopped)

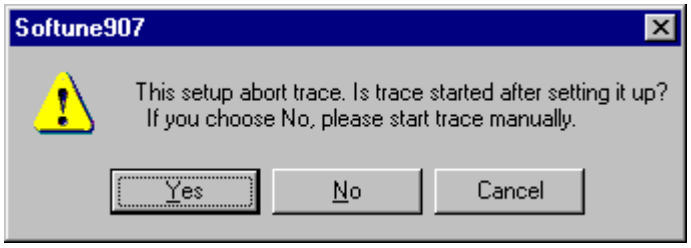


Yes: You can set Filtering or Trace trigger, and it will take effect at once to automatically execute [Start] of the tracing.

No: You can set Filtering or Trace trigger, but you need to manually execute [Start] for it to take effect.

● When tracing

Figure 4.4-36 The setup confirmation dialog when running a user program (During the trace acquisition)



Yes: You can set Filtering or Trace trigger after the automatic [Abort], and it will take effect at once to automatically execute [Start] of the tracing.

No: You can set Filtering or Trace trigger after the automatic [Abort], but you need to manually execute [Start] for it to take effect.

Cancel: Cancels the setting. Select this if you don't want to execute forced stop of tracing.

Notes:

1. Trace controls when a user program is in execution become valid only when the debug function of MB2147-01 is in Trace Enhancement mode. For more information, please refer to Section "2.3.1.6 Debug Function" in SOFTUNE Workbench User's Manual.
2. If you want to use trace controls when a tracing and user program are in execution, you must execute [Abort] of trace acquisition before setting and [Start] of tracing after setting. The changes won't take effect without executing [forced start].
3. "Tracing" means that the trace acquisition is either in "Tracing" or "Pause" state.

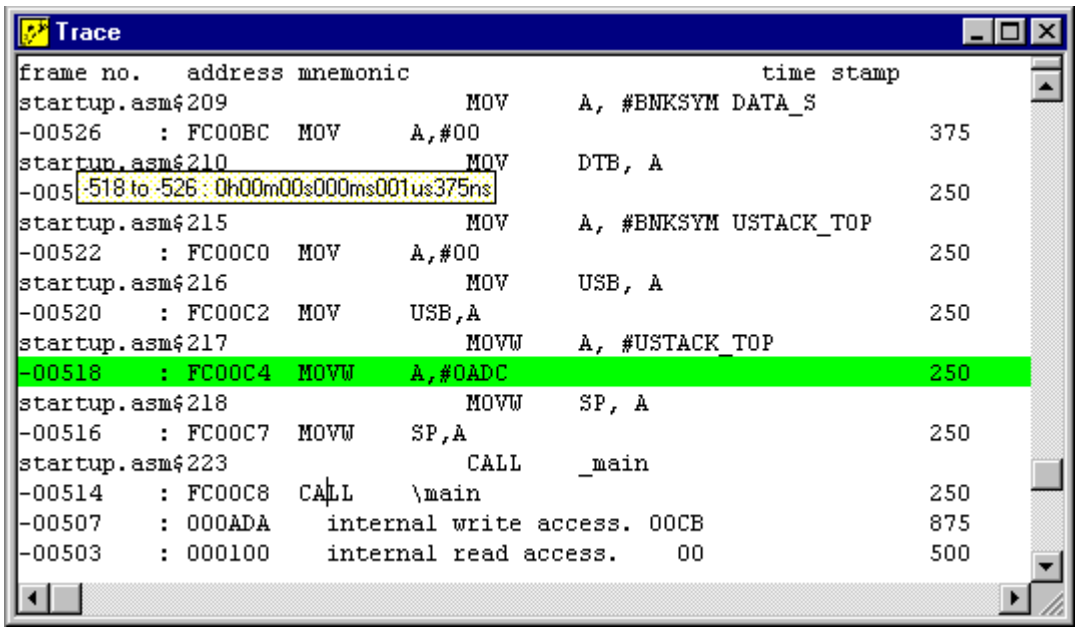
■ Time measurement between frames

You can measure the time between frames in Trace window.

Once you set the frame number with which the time measurement starts, the time between the frame number you set and a frame you are pointing with mouse cursor is popped up.

To set the time measurement start point, select [Time measurement start point] from the shortcut menu on the desired frame number. The frame line will be displayed in green.

Figure 4.4-37 Display of the Time Measurement Result in Trace Window


Trace
frame no. address mnemonic time stamp
startup.asm$209 MOV A, #BNKSYM DATA_S
-00526 : FC00BC MOV A,#00 375
startup.asm$210 MOV DTB, A
-005 -518 to -526 : 0h00m00s000ms001us375ns 250
startup.asm$215 MOV A, #BNKSYM USTACK_TOP
-00522 : FC00C0 MOV A,#00 250
startup.asm$216 MOV USB, A
-00520 : FC00C2 MOV USB,A 250
startup.asm$217 MOVW A, #USTACK_TOP
-00518 : FC00C4 MOVW A,#0ADC 250
startup.asm$218 MOVW SP, A
-00516 : FC00C7 MOVW SP,A 250
startup.asm$223 CALL _main
-00514 : FC00C8 CALL \main 250
-00507 : 000ADA internal write access. 00CB 875
-00503 : 000100 internal read access. 00 500

■ Event Hit Information

You can check whether the events you specified are hit or not.

The events include the following here:

Trace trigger

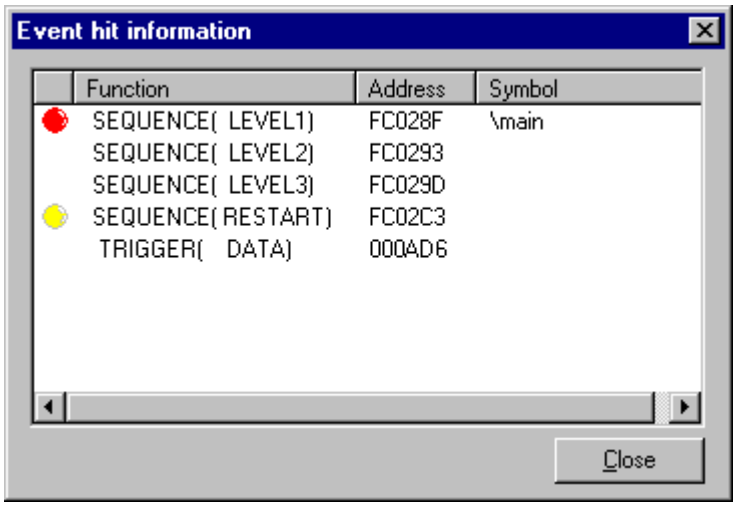
Sequencer

Data watch break

If you have specified some events, select [Event list] from the shortcut menu to check whether any event is hit or not. The event list dialog will appear that includes the events you specified.

When the ❍ of each event starts blinking in 7 colors, which means that there is a hit in the event.

Figure 4.4-38 Event Hit Information Dialog



Note:

Event hit information becomes valid only when the debug function of MB2147-01 is in Trace Enhancement mode. For more information, please refer to Section "2.3.1.6 Debug Function" in SOFTUNE Workbench User's Manual.

4.4.9 Command

"Command" displays the Command Window.

■ Command

A debugger command can be entered and executed directly from the displayed Command Window. The command execution result is also displayed in the Command Window. For the supported debugger commands, refer to "SOFTUNE WORKBENCH Command Reference Manual".

4.4.10 Tool Bar, Status Bar, Tab

Tool bar sets display items. Status Bar or tab switches display to nondisplay or vice versa.

■ Tool bar

Any of the following tool button sets displayed in the tool bar can be selected:

-Common

-Find

-Build

-Debug

-Flag

-Project

For the buttons included in the above sets, see Section "3.2 Tool Bar".

■ Status Bar

"Status Bar" can only switch display to nondisplay or vice versa. For status bar display items, see Section "3.3 Status Bar".

■ Tab

This function enables to choose whether the tab is displayed or non-displayed for switching. By selecting tab function, tab will be attached to the windows. Thus, this makes easy to switch windows.

4.4.11 Object

"Object" displays the Object Window.

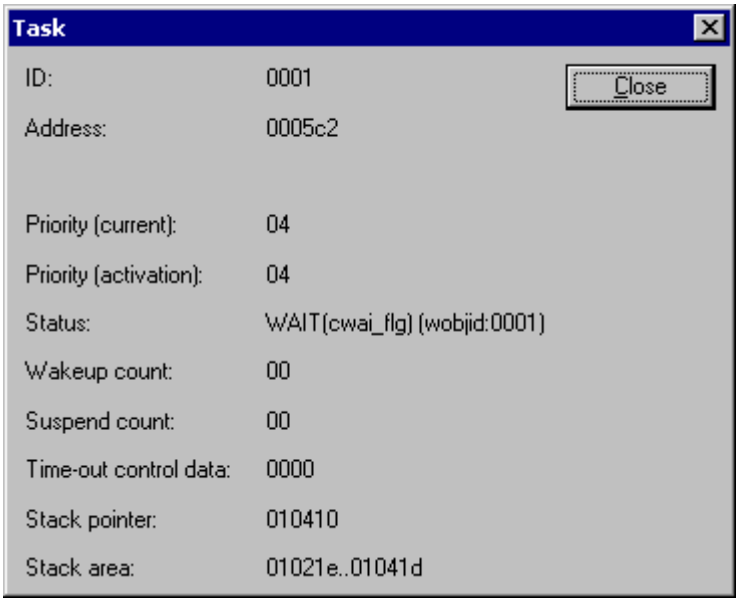
■ Object Window

The Object Window displays the detailed object information used by REALOS. This window can be also used to issue system calls.

To display detailed object information, select an ID, then select [Property] from the shortcut menu.

■ Detailed display of task

Figure 4.4-39 Detailed Task Display Dialog

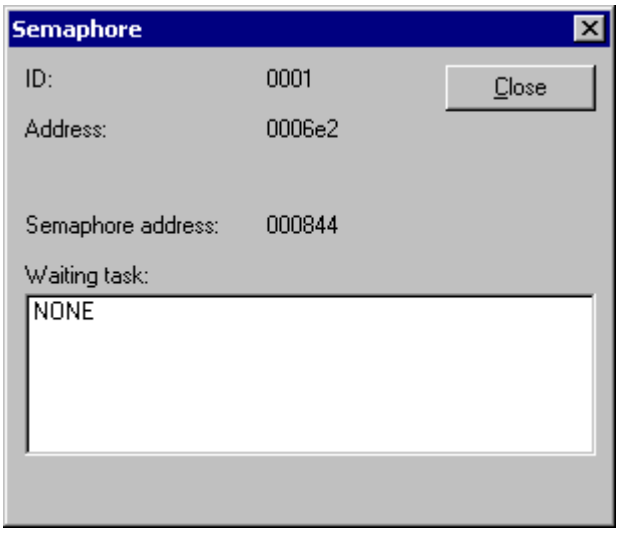


- ID Task	ID number
- Address	Task control block address
- Priority (Current)	Current task priority
- Priority (Start-up)	Task priority at start-up
- Status	Task state
TTS_RDY	Running or ready state
TTS_DMT	Dormant state
TTS_WAI	Wait state
TSS_SUS	Suspend state
Wait factor:	The wait factor is displayed when SOFTUNE WORKBENCH is in the object wait state.
TTW_SLP	slp_tsk/tslp_tsk wait state
TTW_DLY	dly_tsk wait state
TTW_SEM	wai_sem wait state
TTW_FLG	wai_flg wait state

TTW_MBX	rcv_msg wait state
TTW_MPL	get_blk wait state
- Wakeup Count	Wakeup request count
- SUSPEND Count	SUSPEND request count
- Timeout Control Data	Timeout management data
- Stack Pointer	Stack pointer
- Stack Area	Stack area

■ Detailed display of semaphore

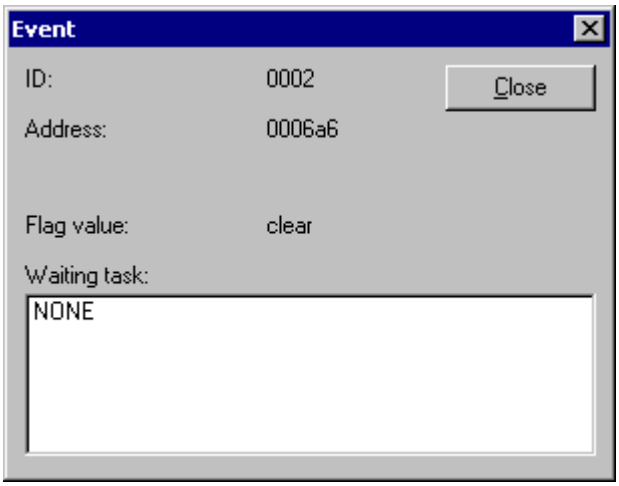
Figure 4.4-40 Detailed Semaphore Display Dialog



- ID	Semaphore ID number
- Address	Address of semaphore control block
- Semaphore Address	Semaphore queue address
- Count	Current semaphore count
- Waiting Task ID	number of task in queue

■ Detailed display of event flag

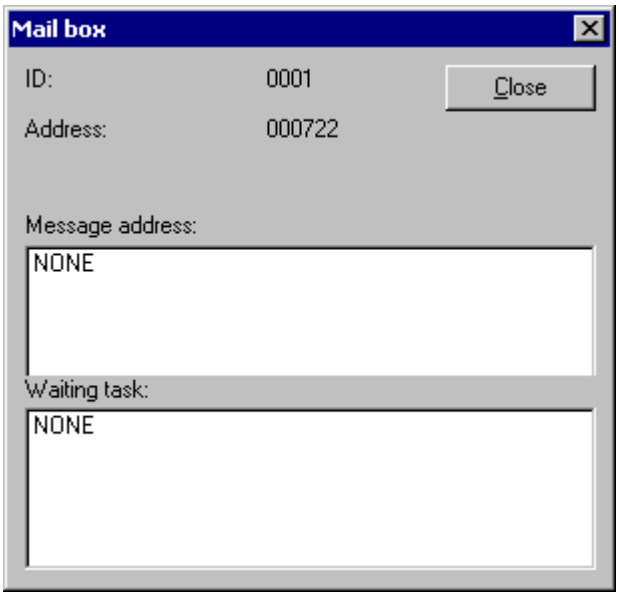
Figure 4.4-41 Detailed Event Flag Display Dialog



- ID event flag ID number
- Address Address of event flag control block
- Flag value current event flag pattern
- Flag address address of current event flag table
- Waiting Task ID number of task in queue

■ Detailed display of mailbox

Figure 4.4-42 Detailed Mailbox Display Dialog

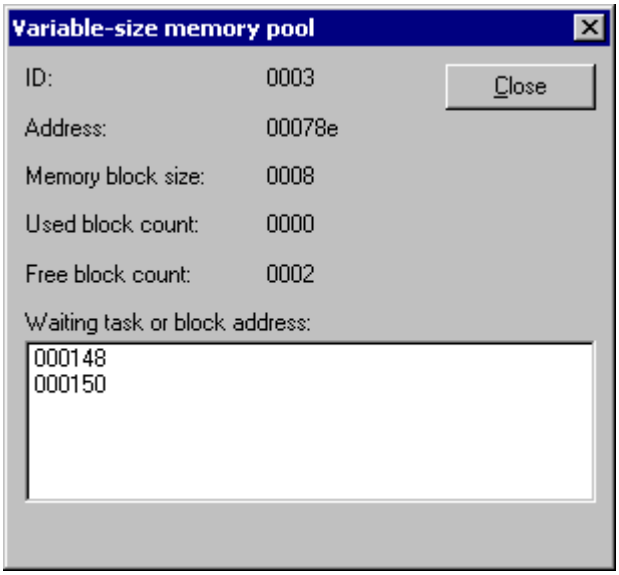


- ID Mailbox ID number
- Address Address of mailbox control block
- Message State Message address
- Waiting Task ID number of task in queue

Waiting Task or Message State is displayed according to the wait factor.

■ Detailed display of fixed-length memory pool

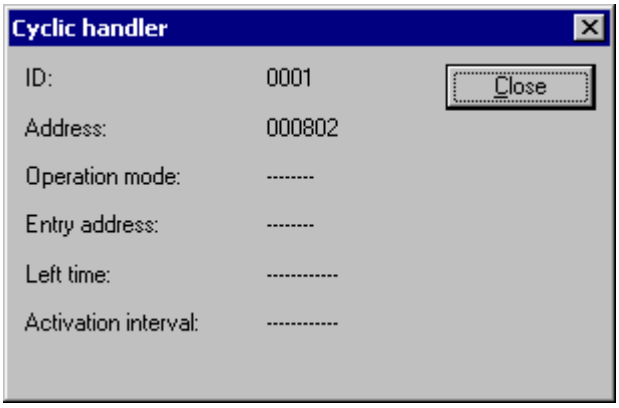
Figure 4.4-43 Detected Memory pool Display Dialog



- ID — ID number of fixed-length memory pool
- Address — Address of fixed-length memory pool control block
- Extended Information — Extended information
- Total Block Count — Count of fixed-length memory blocks
- Block Size — Memory block size
- Free Block Count — Count of free memory blocks
- Waiting Task — Memory pool waiting for task ID

■ Detailed display of cyclic handler

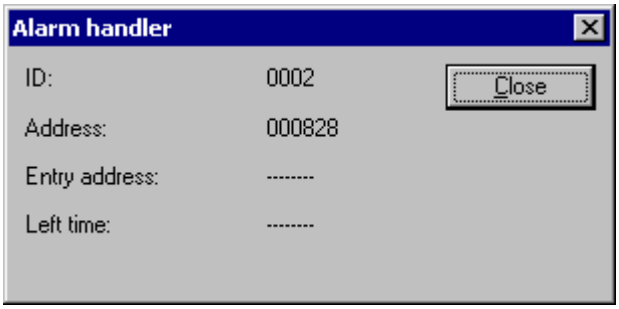
Figure 4.4-44 Detailed Cyclic Handler Display Dialog



- ID — ID number of cyclic handler
- Address — Address of cyclic handler control block
- Operation Mode — Cyclic handler state (Enable/Disable)
- Handler Address — Cyclic handler address
- Left Time — Residual time till activation
- Activation Interval — Cyclic time interval

■ Detailed display of alarm handler

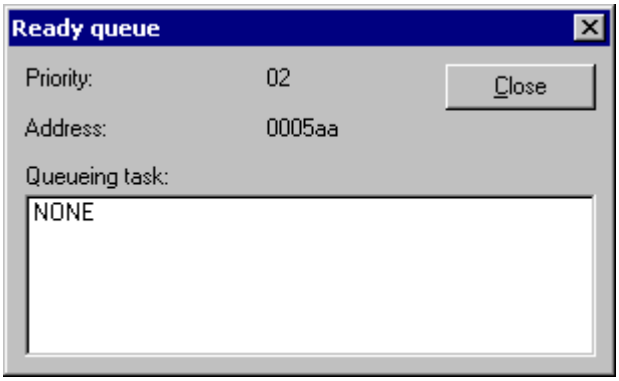
Figure 4.4-45 Detailed Alarm Handler Display Dialog



- ID — Alarm handler number
- Address — Address of alarm handler control block
- Handler Address — Alarm handler address
- Left Time — Residual time till activation

■ Detailed display of ready queue

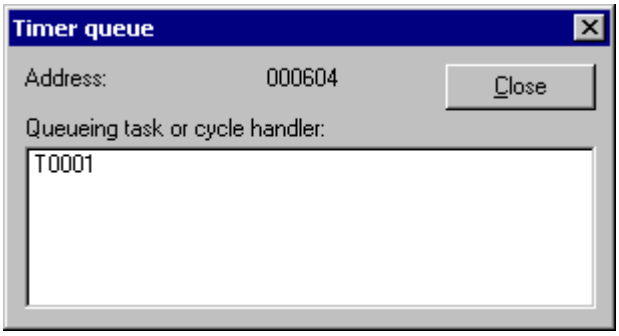
Figure 4.4-46 Detailed Ready Queue Display Dialog



- Priority — Priority
- Address — Address of ready queue header table
- Queued Task ID — ID number of queued task

■ Detailed display of timer queue

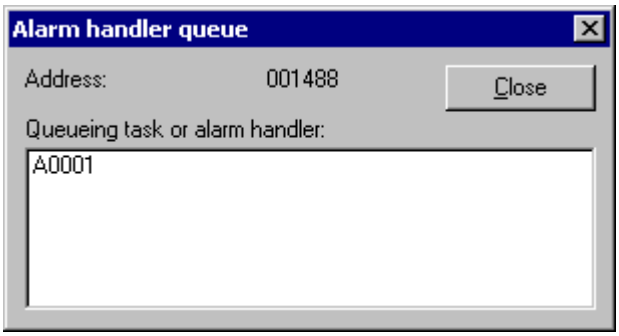
Figure 4.4-47 Detailed Timer Queue Display Dialog



- Address — Timer queue address
- Task ID/handler ID — Cyclic handler ID placed in the timer queue or timeout waiting task ID

■ Detailed display of alarm queue

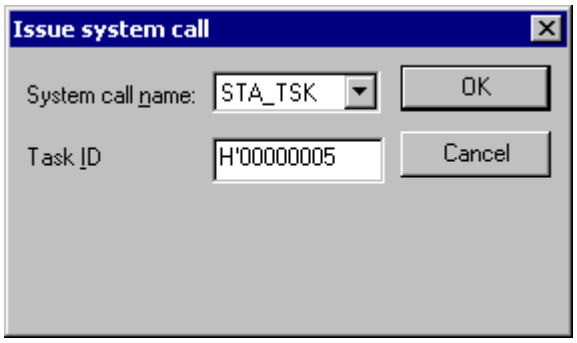
Figure 4.4-48 Detailed Alarm Queue Display Dialog



- Address　　Alarm queue address
- Alarm handler ID　　ID placed in the alarm queue

■ System call issuance

Figure 4.4-49 System Call Insurance Dialog Box



When the system call to be issued is selected from the drop-down list, as shown in the above figure, parameter input for the selected system call dynamically changes. Unnecessary parameters are displayed in gray. Clicking the [OK] button issues the system call. The function call result dialog box opens when the issued system call terminates.

■ System calls that can be issued

- Task

  STA_TSK, TER_TSK, CHG_PRI, SUS_TSK, RSM_TSK, FRSM_TSK, WUP_TSK, CAN_WUP

- Semaphore

  SIG_SEM, PREQ_SEM

- Event flag

  SET_FLG, CLR_FLG

- Mailbox

  SND_MSG, PRCV_MSG

- Memory pool

  PGET_BLK, REL_BLK

- Cyclic handler

  ACT_CYC

4.4.12 Coverage

"Coverage" displays the Coverage Window.

■ Coverage Window

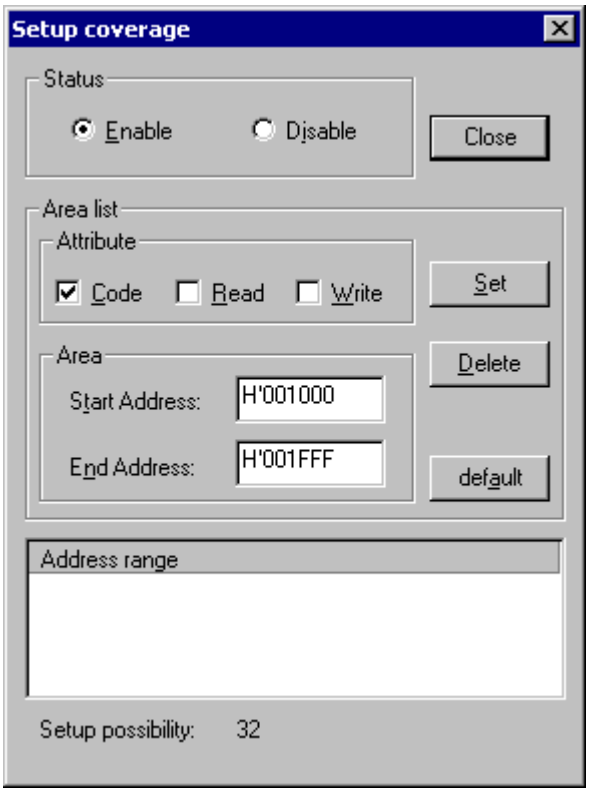
The window displaying the coverage measurement result is opened for setting the coverage and displaying the total coverage rate.

This window can be used only for the debug session for the MB2141/MB2147 (emulator).

■ Coverage setup

Select [Setup] from the shortcut menu of the Coverage Window.

Figure 4.4-50 Coverage Setup Dialog



- State Select [Enable] or [Disable].
- Area list Set [Attribute] and [Area].

To delete an area, click the [Delete] button.

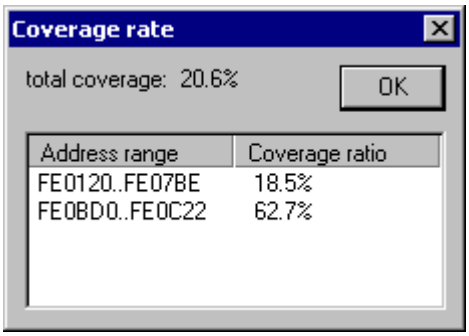
To set an area automatically after the load module file has been read, click [Default].

■ Coverage rate

Select [Total Coverage] from the shortcut menu of the Coverage Window.

The total coverage rate and the coverage rate for each address range are displayed.

Figure 4.4-51 Coverage Rate Dialog Box



4.4.13 Performance

"Performance" displays the Performance Window.

■ Performance Window

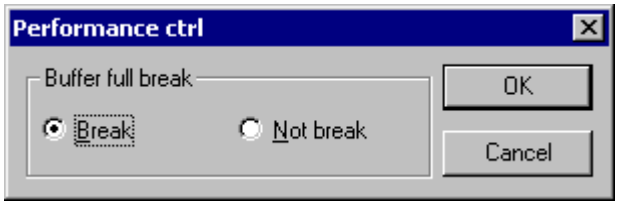
Open the Performance Window to display the performance measurement result. The Performance Window is used to control and set performance.

This function can be used only in the MB2141/MB2147-01 (emulator).

■ Performance control

Select [Setup] from the shortcut menu of the Performance Window. When the buffer is full, set whether to break buffer full.

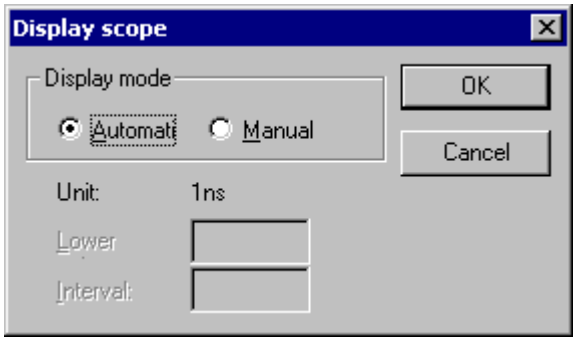
Figure 4.4-52 Performance Control Dialog



■ Performance display setup

Select [Display Scope] from the shortcut menu of the Performance Window.

Figure 4.4-53 Performance Display Setup Dialog



[MB2141]

- Display Mode

  Select [Auto] or [Manual]. When [Manual] is selected, set [Lower Scope] and [Interval Scope].

- Lower Scope

  Specify display start time for detailed measurement result display.

- Interval Scope

  Specify display interval time for detailed measurement result display.

- Unit

  The values set in [Timer minimum measuring unit] of the [Emulation] tab in the debug environment setup dialog box are displayed. This dialog opens when [Debug environment] is selected from the [Setup] Menu.

[MB2147-01]

The Measurement unit is fixed at 1ns.

■ Copy of the Performance Measurement Result

The measurement result displayed in the performance window can be copied to a clipboard. To do this, select [Copy] from the shortcut menu while the measurement result is displayed in the performance window.

[Examples]

When the display type of MB2147-01 is "measurement time"

```
event    = 1 -> 2
min time = 7.050
max time = 26.225
avr time = 13.763
---------------------------------------
            time (us)         |   count
------------------------------+---------
        0.000 -         7.999 |       1
        8.000 -         8.999 |       0
        9.000 -         9.999 |       0
       10.000 -        10.999 |       2
       11.000 -        11.999 |       0
       12.000 -        12.999 |       1
       13.000 -        13.999 |       1
       14.000 -        14.999 |       2
       15.000 -        15.999 |       0
       16.000 -        16.999 |       0
       17.000 -        17.999 |       0
       18.000 -               |       1
------------------------------+---------
            total             |       8
```

When the display type of MB2147-01 is "measurement time"

```
no.   count
1         8
2         8
3        15
4        26
```

Note:

The contents output to a clipboard by [Copy] form the shortcut menu are same as those output by SHOW PERFORMANCE command. For more information, please refer to Section "4.8 SHOW PERFORMANCE" in SOFTUNE Workbench Command Reference Manual.

4.4.14 Fonts

The Fonts for each window are changed.

■ Setting fonts

The font information (font name and size) currently set for each window can be displayed and the setting can be changed. Also, all font settings can be set to the defaults.

During debugging, the fonts for debug-related windows (such as Source window) can be changed.

■ Changing fonts

Change fonts as follows:

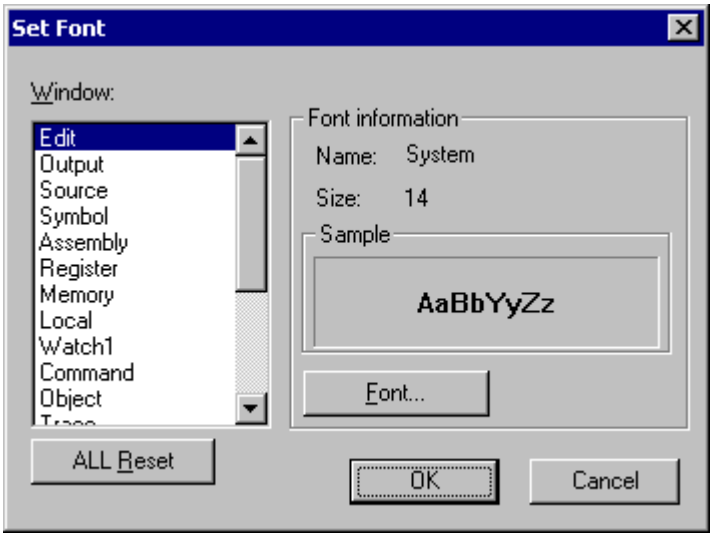
1. Select the [Display]-[Font] menu.
   - The font setting dialog is displayed (Figure 4.4-54).
2. Select the window with the font to be changed and click the [Font...] button.
   - The font setting dialog is displayed (Figure 4.4-54).
3. Specify the font name and size and click the [OK] button. The font type that can be selected depends on the window.
4. When the [OK] button in the font setting dialog (Figure 4.4-54) is clicked, the window fonts are changed.

■ Resetting fonts

Reset fonts as follows:

1. Select the [Display]-[Font] menu.
   - The font setting dialog is displayed (Figure 4.4-54).
2. Click the [All Reset] button.
3. The fonts for the window displayed in [Window] are all reset to the defaults.
4. When the [OK] button is clicked, the window fonts are returned to the defaults.

Figure 4.4-54 Set Font Dialog



4.4.15 Real-time Memory

The real-time memory window is displayed.

■ Real-time memory window

The window to display the mirror memory is opened. The memory contents are dumped (displayed) from the specified address.

This window can be only used for the debug session for the MB2147-01 (emulator).

The location accessed by the user program is color-coded. The value which is not accessed may be different from that of real memory. (The value of memory accessed by the user program is only valid, but that of memory accessed by operating debugger is not valid.)

- Invalid memory contents

  The contents of the displayed mirror memory are undefined (displayed in grey).

- Valid memory contents

  The contents of the displayed mirror memory are same as those of the real memory (displayed in black or red).

  Only memory contents at locations accessed at execution are valid and they are displayed in black. Locations where memory contents are changed are displayed in red.

- When memory window has been already opened

  This window is activated.

- When memory window has not been yet

  The area specified using [Setup] - [Debug Environment] - [Realtime memory Area] is displayed.

■ Area

To move the display area for the real-time memory window, click the right mouse button on the real-time memory window to display the menu, and then specify [Area] and select [Area 1] or [Area 2]. Then, the area specified using [Setup] - [Debug Environment] - [Realtime memory Area] is displayed.

■ Display setting

To set the display format for the real-time memory window, click the right mouse button on the real-time memory window to display the menu, and then select [Setup] to display the setting dialog (Figure 4.4-55).

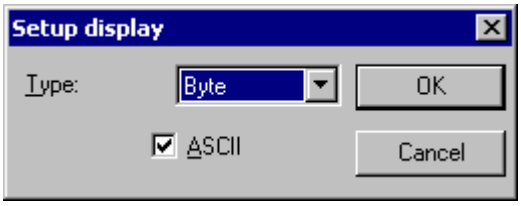
- Display format

  The display format for the window is specified.

  Select from Bit, Byte, Word, and Long.

  Whether or not to display ASCII characters on the right side of the window is selected.

Figure 4.4-55 Display Setting Dialog



■ Area setting

To set the display area for the real-time memory window, click the right mouse button on the real-time memory window to display the menu, and then select [Setup Area] to display the [Setup] - [Debug Environment] - [Realtime memory Area] menu.

Note:

Coverage function should be disabled for "0" bank monitoring. For more information, see Section "2.3.9 Real-time Memory" in SOFTUNE Workbench User's Manual.

4.4.16 RAM Checker

The RAM Checker window is displayed.

■ RAM Checker

Open the RAM Checker window to display the logging status and the monitoring of the monitoring address. In addition, the shortcut menus can be used to set the monitoring address and turn-on/off of logging.

4.5 Project

"Project" sets project-related items and executes make/build.

■ Project-related item setup

- Active Project
- Add Project
- Add Member
- Setup Workspace
- Setup Project
- Setting Customize Build
- Project Dependencies
- Project Configuration
- Include Dependencies

■ Make/build execution

- Compile
- Make
- Build
- Stop

Note:

No selection can be made during debugger startup. Quit the debugger.

4.5.1 Active Project

The active project is replaced.

■ Active Project

When the names of all projects in workspace are displayed in the submenu, click the name of the project that is made active.

A checkmark is placed at the left side of the active project name in the menu.

4.5.2 Add Project

A project is added to workspace.

■ Add Project

There are the following two menus to add a project:

- New

  A new project is created and stored in workspace.

- Project

  An existing project is stored in workspace.

Note:

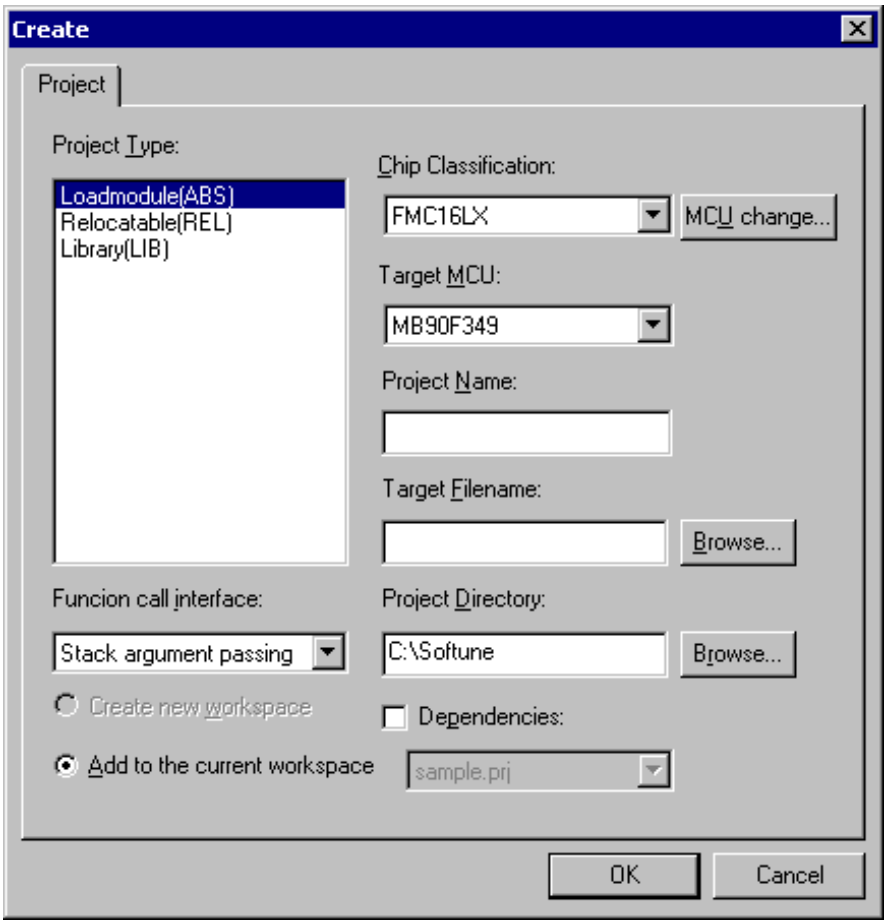
If any project having the same name as that of the project to be stored is in workspace, it cannot be stored in workspace.

4.5.2.1 Add Project - Create

A new project is created and stored in workspace.

■ Add Project - Create

Figure 4.5-1 Create Project Dialog



- Project Type

  Select the project type. For details about the project type, see Section "2.4 Storing Project".

- Create new workspace

  New workspace is created for storage of a project. This item cannot be selected.

- Add to the current workspace

  A project is stored in the currently opened workspace.

- Chip Classification

  Selects the chip classification.

- Target MCU

  Selects the target MCU.

- Project Name

  Specifies the project name.

- Dependencies

When the project name is specified in the combo box, the project to be stored is defined as a subproject in the specified project.

■ Procedure for Addition

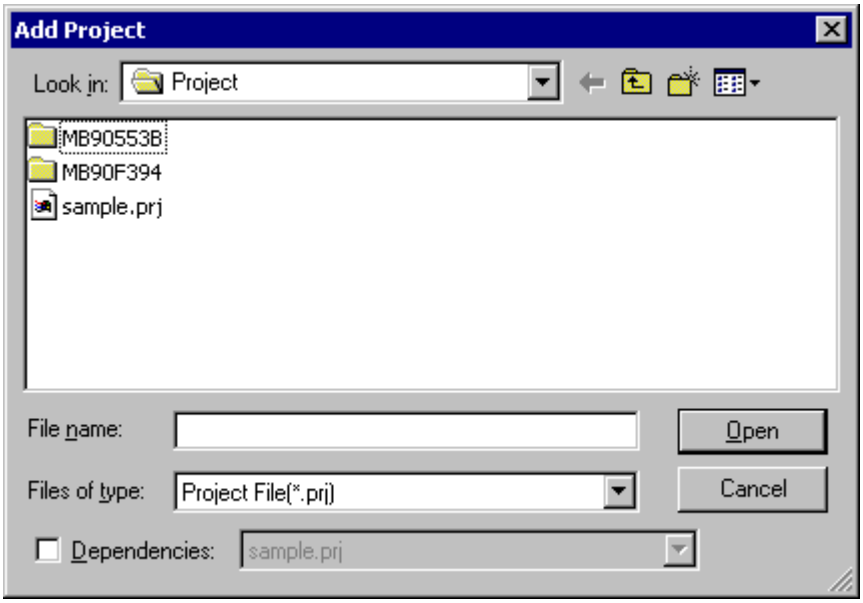
For the procedure for addition of a project, see Section "2.4 Storing Project".

4.5.2.2 Add Project - Existing Project

An existing project is stored in workspace.

■ Add Project - Existing Project

Figure 4.5-2 Add Project Dialog



- Dependencies

  When the project name is specified in the combo box, the project to be stored is defined as a subproject in the specified project.

■ Procedure for Addition

For the procedure for addition of a project, see Section "2.4 Storing Project".

4.5.3 Add Member

"Add Member" adds a file to the project.

■ Add Member

There are the following menus to add a member:

- File

  A file is specified and stored in the project.

- Directory

  A directory is specified to store its file and folder in the project.

  The file dialog box for file selection is displayed. Two or more files can be selected at a time.

  The selected files are added to the project and displayed in the Project Window.

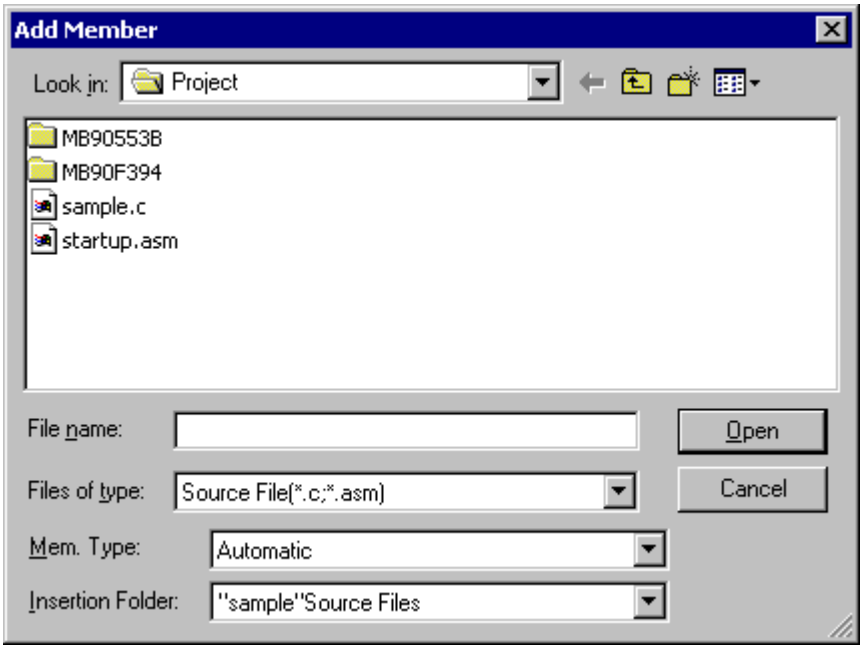
4.5.3.1 Add Member - File

A file is specified and stored in the project.

■ Add Member - File

The file dialog box for file selection is displayed. Two or more files can be selected at a time.

Figure 4.5-3 Add Member Dialog-File



- Mem. Type

  The file to be stored is stored as the type of specified member in the project. If "Auto" is selected, the type of member is determined by the extension.

- Insertion Folder

  The project in which a file is stored and the folder into which the file is inserted are specified. The character string enclosed between "and" is the project name.

■ Procedure for Addition

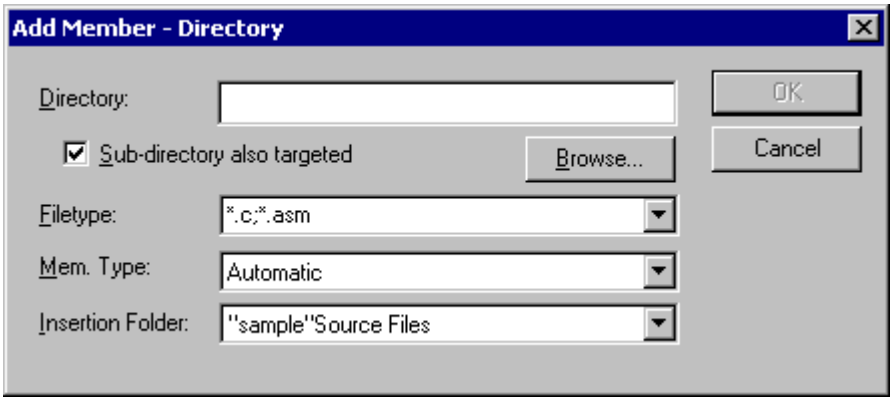
For the procedure for addition of a member, see Section "2.5 Creating and Registering Source File in Project".

4.5.3.2 Add Member - Directory

A directory is specified to store its file and folder in the project.

■ Add Member - Directory

Figure 4.5-4 Add Member Dialog-Directory



- Directory

  Specifies the directory having the file to be stored.

- Sub-directory also targeted

  When turned on, the file in the subdirectory in the specified directory is stored. A subdirectory is hierarchically created as a folder in the project.

- File type

  Only the file having a specified extension is stored in the project.

- Mem. Type

  The file is stored as the type of the specified member in the project. If "Auto" is selected, the type of member is determined by the extension.

- Insertion Folder

  Specify the project in which a file is stored and the folder into which the file is inserted. The character string enclosed between "and" is the project name.

■ Procedure for Addition

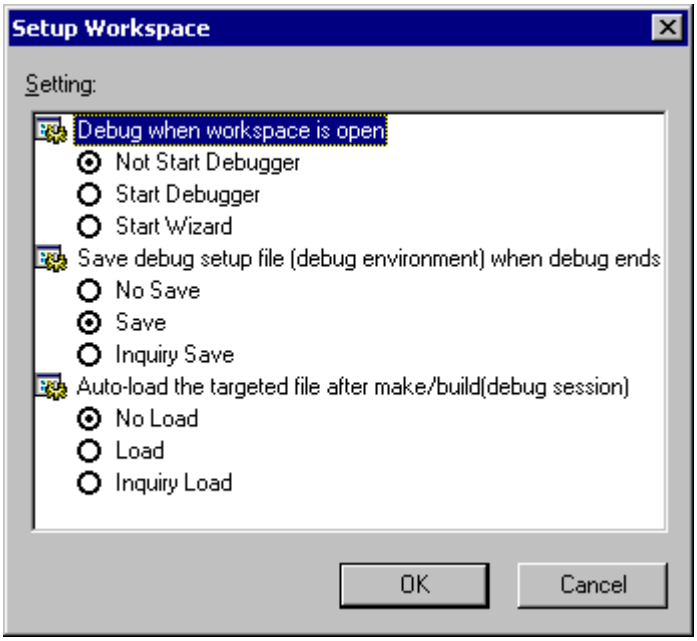
For the procedure for addition of a member, see Section "2.5 Creating and Registering Source File in Project".

4.5.4 Setup Workspace

The basic setting regarding workspace is performed.

■ Setup Workspace

Figure 4.5-5 Setup Workspace Dialog



- Debug when workspace is open

  Specifies the debug action just after the workspace opens.
  (Not Start Debugger/Start Debugger/Start Wizard)

- Save debug setup file (debug environment) when debug ends

  Specifies whether to save setup information upon completion of debug session.
  (No Save/Save/Inquiry Save)

- Auto-load the targeted file after make/build (debug session)

  Specifies whether to reload target file after make/build executed in debug session.
  (No Load/Load/Inquiry Load)

4.5.5 Setup Project

Setting regarding the project is performed.

■ Setup Project

The Setup Project dialog has the part where the target item is set and the part where setting is performed.

The set values are enabled when the [Apply] or [OK] button is clicked.

The directory can be described in a relative path from the project.

If two or more target items are set, the values are displayed as follows:

- Edit

  If the values are equal in all target items, they are displayed as they are.

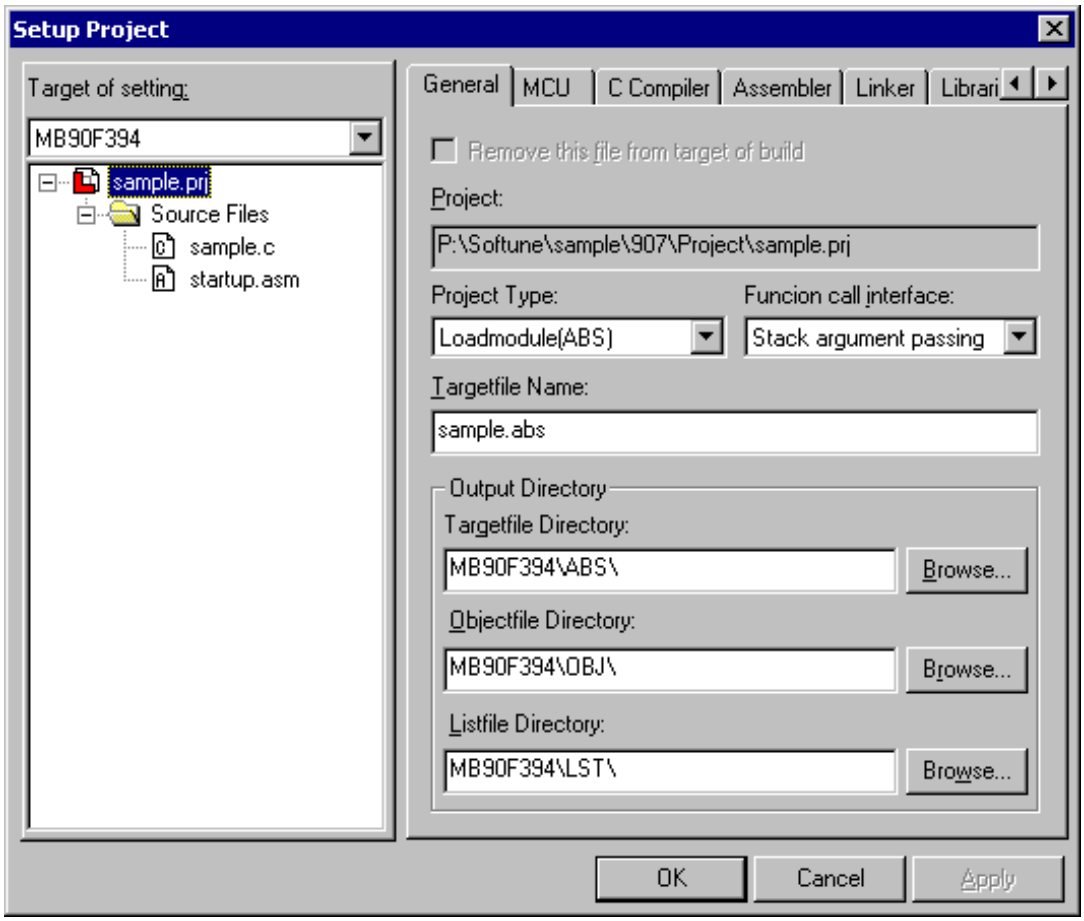
  If the values are not equal in at least one target item, they are blanked.

- Check button

  If the values are equal in all target items, they are displayed as they are.

  If the values are not equal in at least one target item, they are dimmed.

Figure 4.5-6 Setup Project Dialog



■ Setup Target

The items to be set in the combo box and tree view at the left side of the dialog are specified.

- Target of setting combo box:

All the configuration names of projects in workspace are displayed.

Configuration name: The selected configuration is set.

[All Configuration]: All configurations are set.

[Multiple Configuration]: The Multiple Configuration dialog (Figure 4.5-7) is opened. The two or more configurations specified in the dialog are set.

- Target of setting tree view:

All projects having the configuration names specified in the combo box are displayed. The items that can be set vary depending on the selected items. When two or more items are selected, they can be changed at a time. If the items of different types are selected, only overlapped items can be set.

Project: The [General] items, [MCU] items, common options ([C Compiler], [Assembler], [Linker], [Librarian]), and [Debug] items can be set.

C source file: The [General] items and individual options ([C Compiler]) can be set.

Assembler source file: The [General] items and individual options ([Assembler]) can be set.

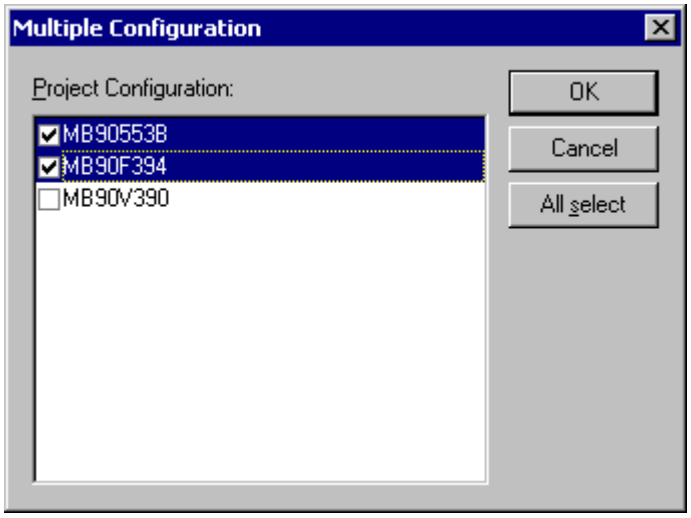
Library file: The [General] items can be set.

Object file: The [General] items can be set.

Relative file: The [General] items can be set.

Folder: All files in the folder are set.

Figure 4.5-7 Multiple Configuration Dialog



■ Setting Items

The following items can be set or changed in tab form.

- General

  The project type, target name, output directory, and file building are set.

- MCU

  The items regarding the MCU, such as the chip type and target MCU, are set.

- C Compiler

  The C compiler options are set.

- Assembler

  The assembler options are set.

- Linker

  The linker options are set.

- Librarian

  The librarian options are set.

- Converter

  The converter start and converter options are set.

- Debug

  The debug options and debug setup information are set.

4.5.5.1 General

This section explains the items that can generally be set in the [Setup Project] dialog.

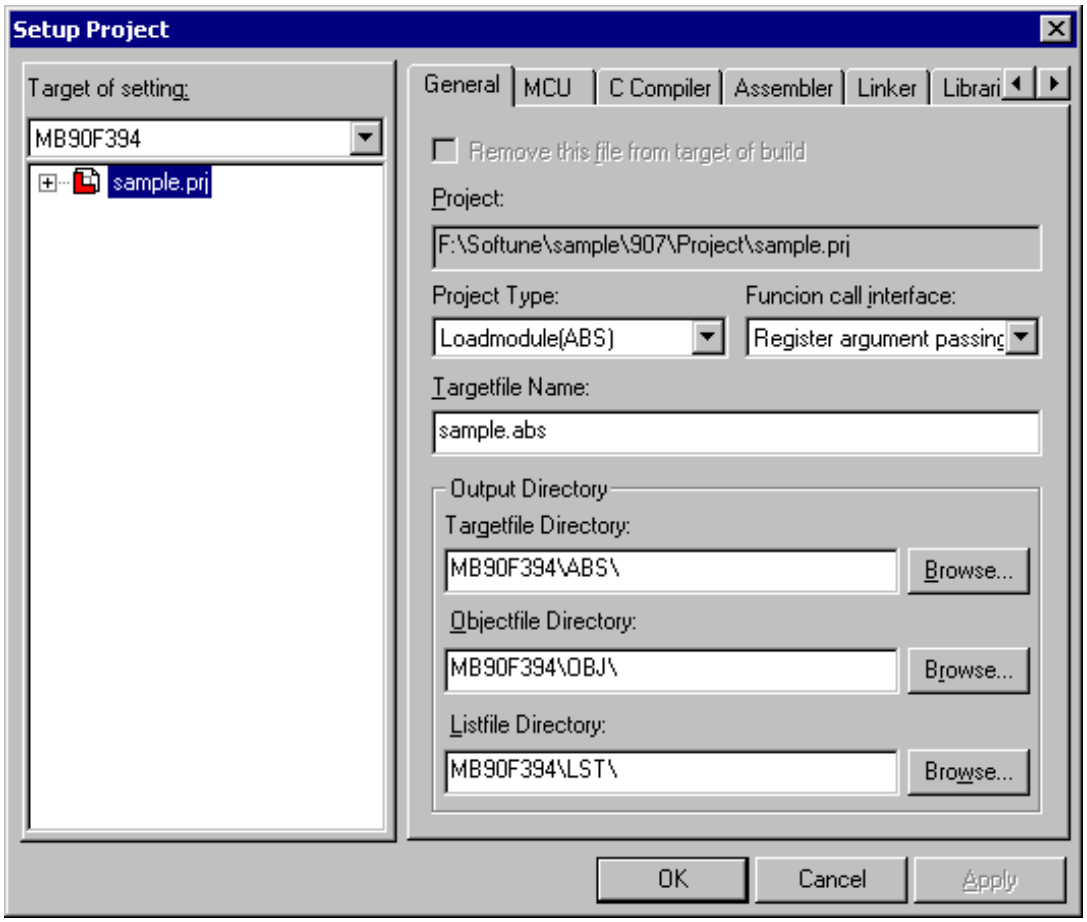
■ Setting of General

1. Click the [General] tab in the [Setup Project] dialog.
2. Set the following items as needed:
   [Remove this file from target of Build], [Project type], [Output Directory], [Target file Name]

[Remove this file from target of Build], [Project type], [Function call interface] or [Output Directory] can be selected as category.

Specified options can be checked using [Option] at the bottom of the dialog.

Figure 4.5-8 Setup Project Dialog-General



- Remove this file from target of build

  The file selected in the tree view is removed from the Make/Build target. This item can be set when C source file, Assembler source file, Library file, Object file, or Relative file is selected.

- Project

  The full path to the project file is displayed.

- Project Type

  Set the type of project file selected in the tree view (absolute (ABS)/relative (REL)/library (LIB)). If the REALOS (ABS)) type is selected, the project type cannot be changed. This item can be set when the project file is selected singly.

  The project type cannot be set for each configuration.

- Function-call Interface

  Select the function-call interface to be used in C Compiler.

  Table 4.5-1 shows selectable function-call interfaces and their descriptions.

  The function-call interface cannot be set for each configuration.

- Target File Name

  Set the target file name.

- Target File Directory

  Set the directory of the target file. This item can be set when only Project (two or more items may be set) is selected.

- Object File Directory

  Set the output directory of the object file output in compiling or assembling. This item can be set when only Project (two or more items may be set) is selected.

- List File Directory

  Set the output directory of the list file output in compiling or assembling. This item can be set when only Project (two or more items may be set) is selected.

Table 4.5-1 List of function-call interfaces

Function-call interfaces	Explanation
arguments by the stack	Passes function arguments by the stack. This is a default setting
arguments by registers	Passes selected function arguments by registers.These processings reduce the code size and stack usage to improve the program execution speed.

Note:

For details of the function-call interface, see "Appendix H Guide to Changing Function-call Interface."

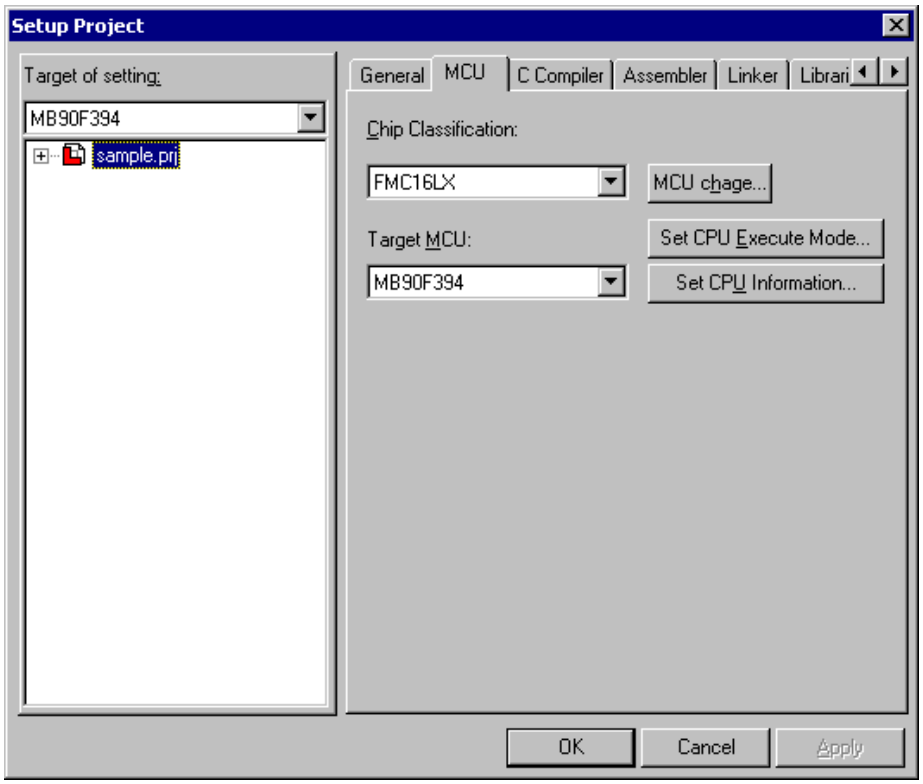
4.5.5.2 MCU

This section explains the items that can be set generally in the [Setup Project] dialog.

■ Setting of MCU

1. Click the [MCU] tab in the [Setup Project] dialog.
2. Set the following items as needed: [Chip Classification], [Target MCU], [Set CPU Execute Mode], [Set CPU information].

Figure 4.5-9 Setup Project Dialog-MCU



- Chip Classification

  Selects the type of chip.

- Target MCU

  Select the target MCU of the chip type selected in [Chip Classification].

- MCU change

  Opens MCU change dialog box.

- Set CPU Execute Mode

  The Set CPU Operation dialog is opened.

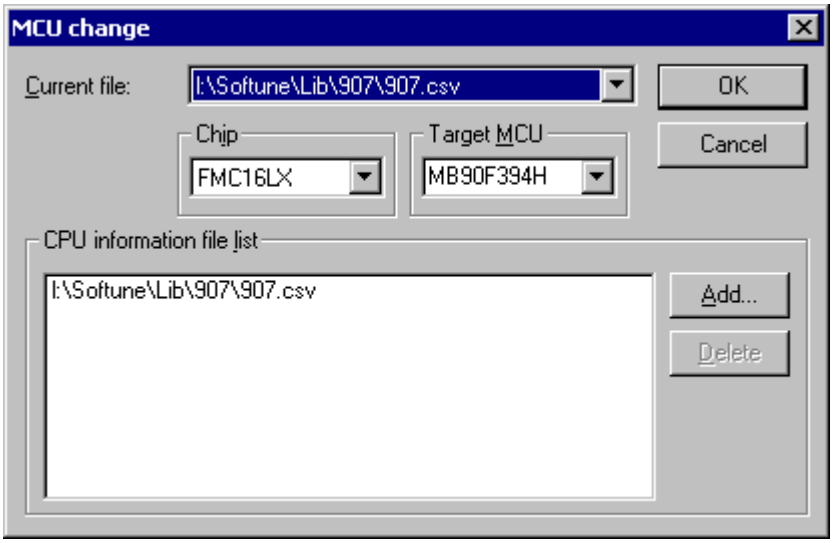
- Set CPU Information

  The Set CPU Information dialog is opened.

■ Setting of MCU change dialog box

1. Click [MCU change] button.
   The MCU change dialog box shown in Figure 4.5-10 opens.
2. From [Current file], choose the CPU information to be used. If a CPU information file to be used is not found, add it with the [Add] button for [CPU information file list].
   The MCU list for the selected CPU information file appears in [Chip] and [Target MCU].
3. Choose the MCU to be changed from the [Chip] and [Target MCU], and click [OK] button.
   The change you specified takes effect.

Figure 4.5-10 MCU change dialog box



- Current file

  Choose the CPU information file to be used. The MCU list for the selected CPU information file appears in [Chip] and [Target MCU].

- Chip

  Choose a chip type.

- Target MCU

  Choose a target MCU for the chip type that is chosen from the [Chip] area.

- CPU information file list

  Lists the registered CPU information files.

- Add

  Adds a CPU information file.

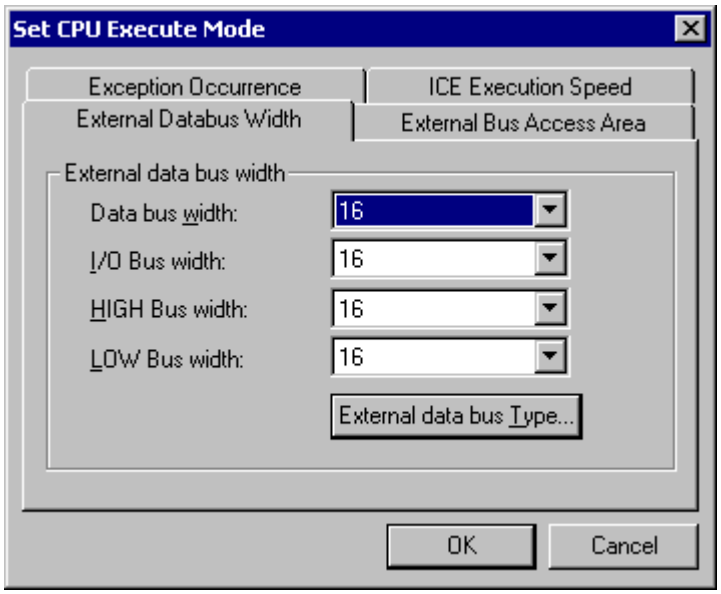
- Delete

  Deletes a CPU information file that is chosen from the [CPU information file list].

  Note that the files that are retrieved by default cannot be deleted.

- CPU Execution Operation Setup - External Data Bus Width

Figure 4.5-11 CPU Execution Operation Setup Dialog (External Data BUS Width)



- Data Bus Width

  Specifies external data bus width.

- I/O Bus Width

  Specifies I/O bus width.

- HIGH Bus Width

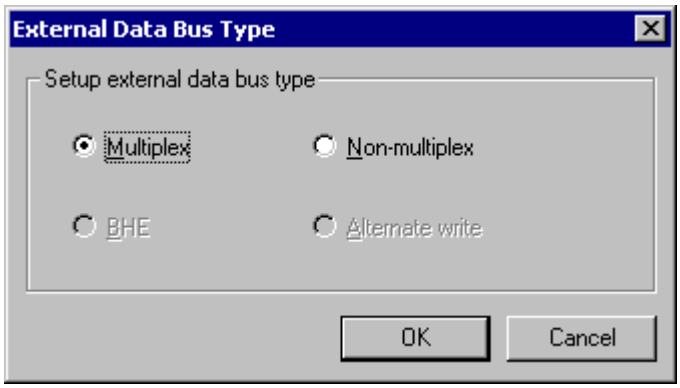
  Specifies HIGH bus width.

- LOW Bus Width

  Specifies LOW bus width.

- External Data Bus Type

Figure 4.5-12

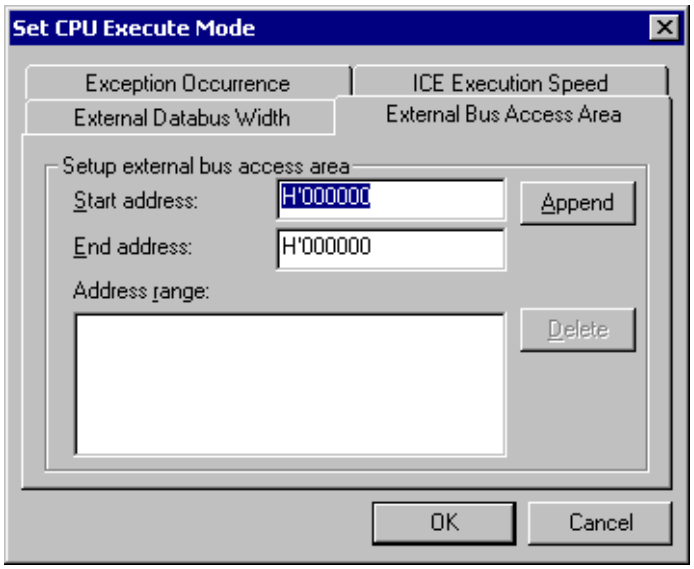


- External Data Bus Type Setup

  Sets external data bus type.

- CPU Execution Operation Setup - External Bus Access Area

Figure 4.5-13 CPU Execution Operation Setup Dialog (External Bus Access Area)



- Start Address

  Specifies start address of external bus access area.

- End Address

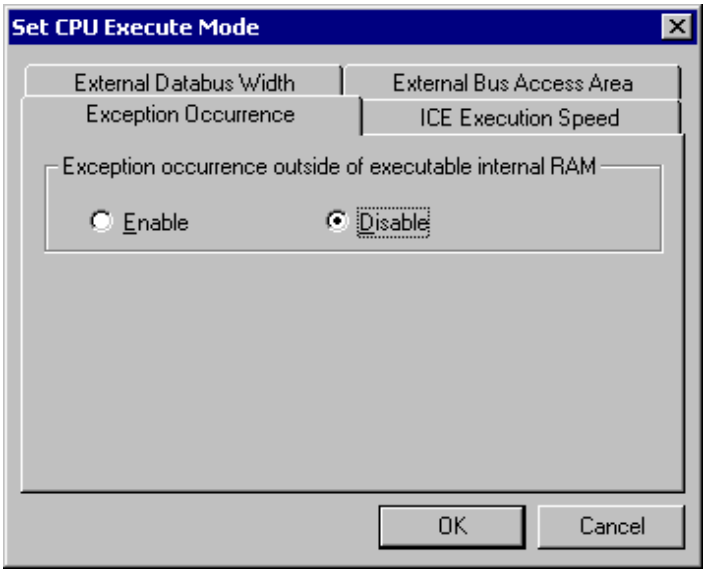
  Specifies end address of external bus access area.

- Address Range

  Specifies address range list of external bus access area.

- CPU Execution Operation Setup - Exception Occurrence

Figure 4.5-14 CPU Execution Operation Setup Dialog (Exception Occurrence)

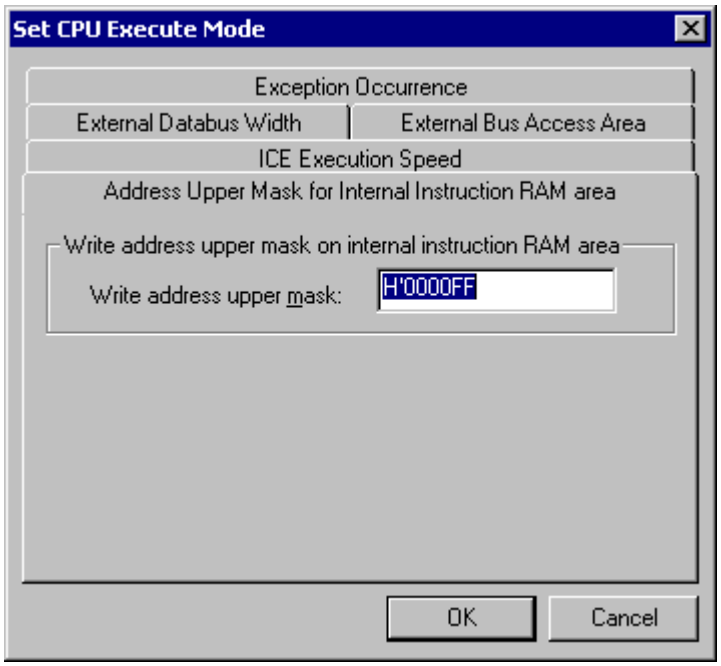


- Exception Occurrence Outside of Executable Internal RAM

  Sets CPU execution operation when exception occurs during execution of internal RAM area (Enable/Disable).

- CPU Execution Operation Setup - Upper mask for Internal Instruction RAM

Figure 4.5-15 CPU Execution Operation Setup Dialog (High-Order Mask Internal Instruction RAM Area)

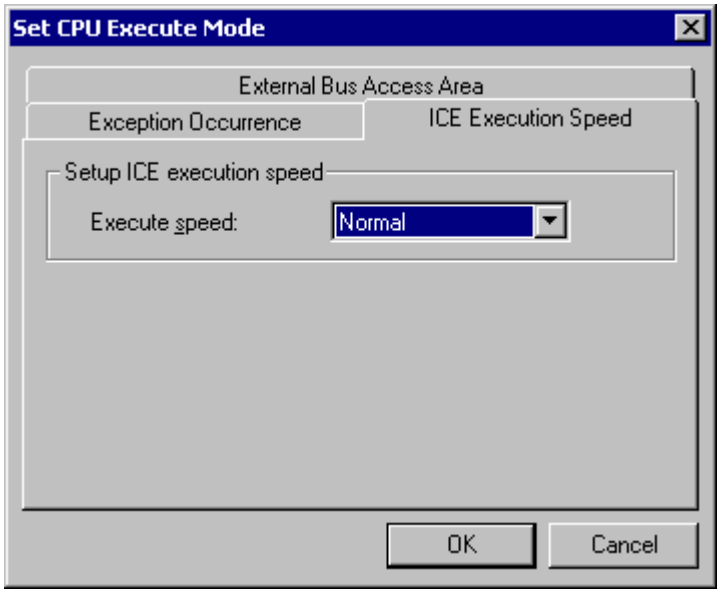


- Write Address Upper Mask

  Specifies high-order write address mask in external instruction RAM area.

- CPU Execution Operation Setup - ICE Execute Speed

Figure 4.5-16 CPU Execution Operation Setup Dialog (ICE Operation Speed)



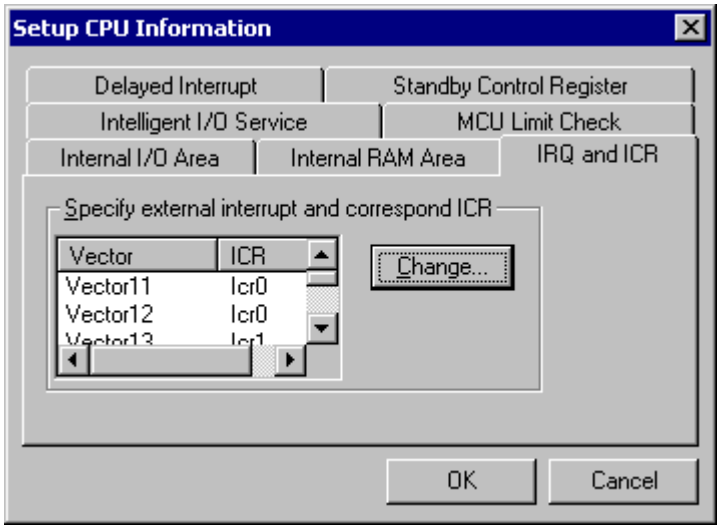
- Execute Speed

  Sets ICE operation speed (High Speed/Normal).

■ Setting of CPU Information

- Setup CPU information - IRQ and ICR

Figure 4.5-17 CPU Information setup Dialog (IPQ and ICR)



- Relationship between Vector and ICR

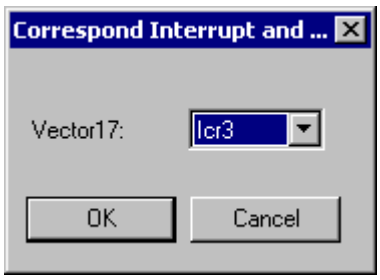
The correspondence between external interrupts (Vector) and the interrupt control register (ICR) varies with the target MCU.

This correspondence is specified by the simulator debugger.

- Setting Correspondence between External Interrupts and ICRs

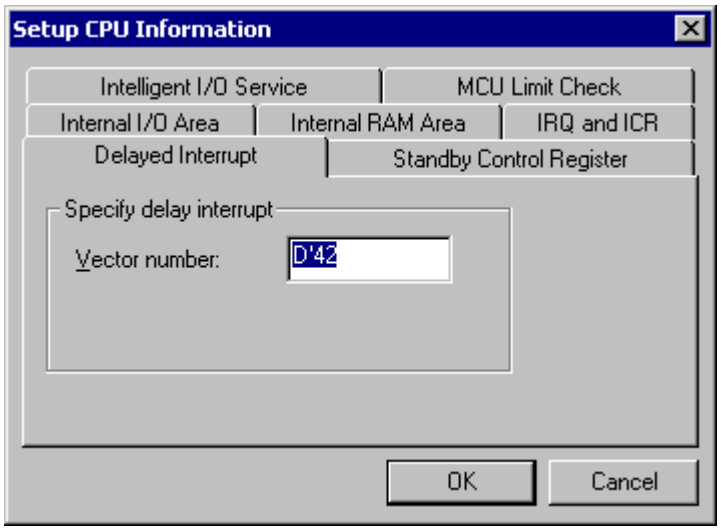
Set the interrupt control registers (ICRs) corresponding to vector numbers. When a vector number is selected and the [Change] button is clicked, the dialog shown in Figure 4.5-18 opens.

Figure 4.5-18 Correspondence between External Interrupts and ICRs



- Setup CPU Information - Delayed Interrupt

Figure 4.5-19 CPU Information Setup Dialog (Delayed Interrupt)

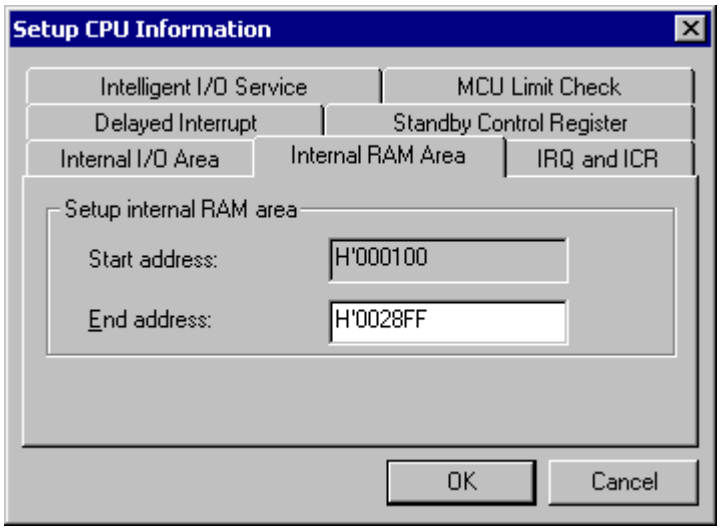


- Vector Number

  Sets vector number of delayed interrupt.

- Setup CPU Information - Internal RAM Area

Figure 4.5-20 CPU Information Setup Dialog (Internal RAM Area)



- Start Address

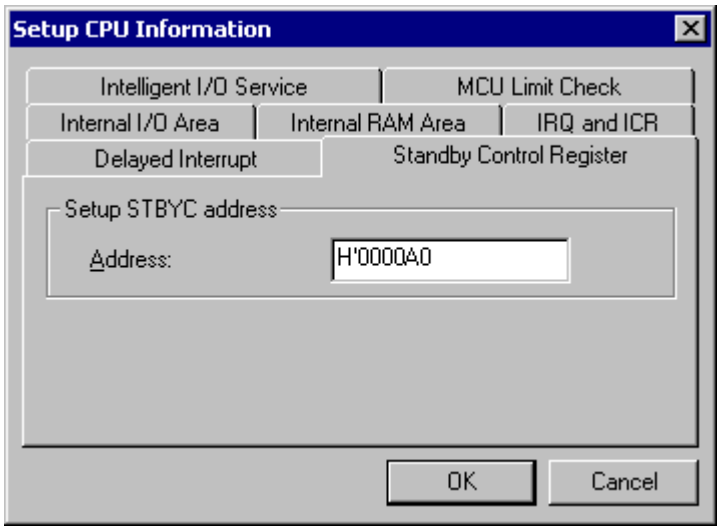
  Sets start address of internal RAM area.

- End Address

  Sets end address of internal RAM area.

- Setup CPU Information - Standby Control Register

Figure 4.5-21 CPU Information Setup Dialog (Standby Control Register)

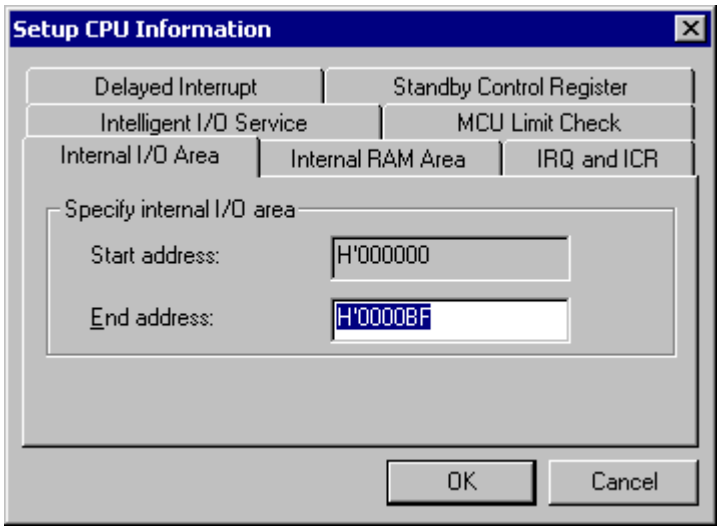


- Address

  Sets STBYC address.

- Setup CPU Information - Internal I/O Area

Figure 4.5-22 CPU Information Setup Dialog (Internal I/O Area)



- Start Address

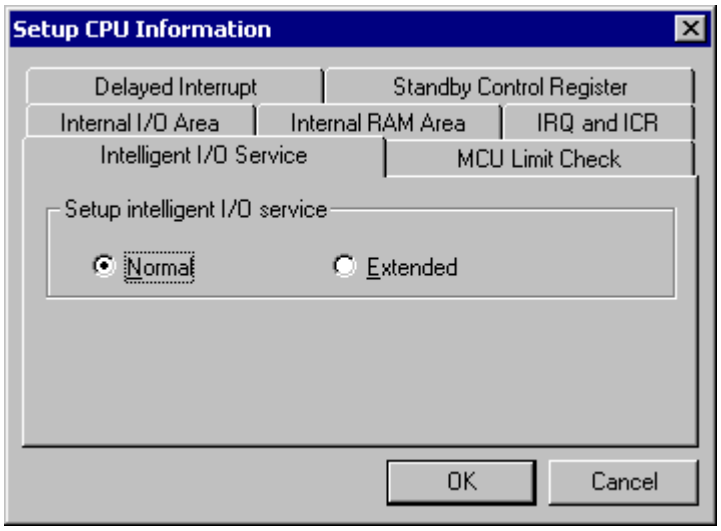
  Sets start address of internal I/O area.

- End Address

  Sets end address of internal I/O area.

- Setup CPU Information - Intelligent I/O Service

Figure 4.5-23 CPU Information Setup Dialog (Intelligent I/O Service)

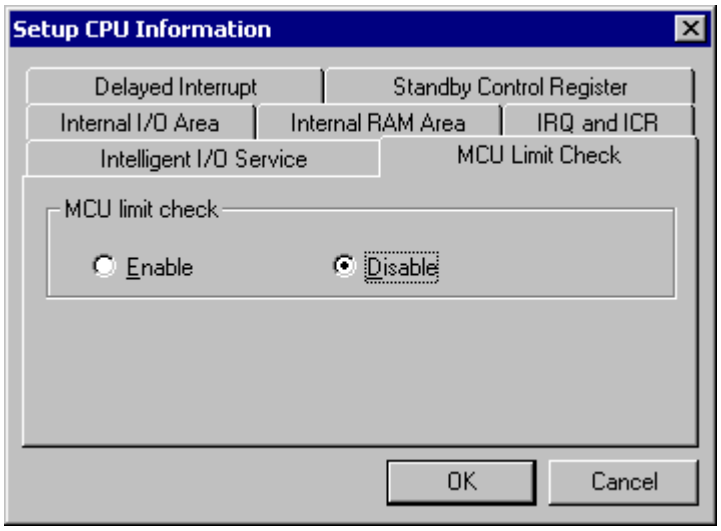


- Intelligent I/O Service Setup

  Sets intelligent I/O service (Normal/Extension).

- Setup CPU Information MCU Limit check

Figure 4.5-24 CPU Information Setup Dialog (MCU Limit Check)

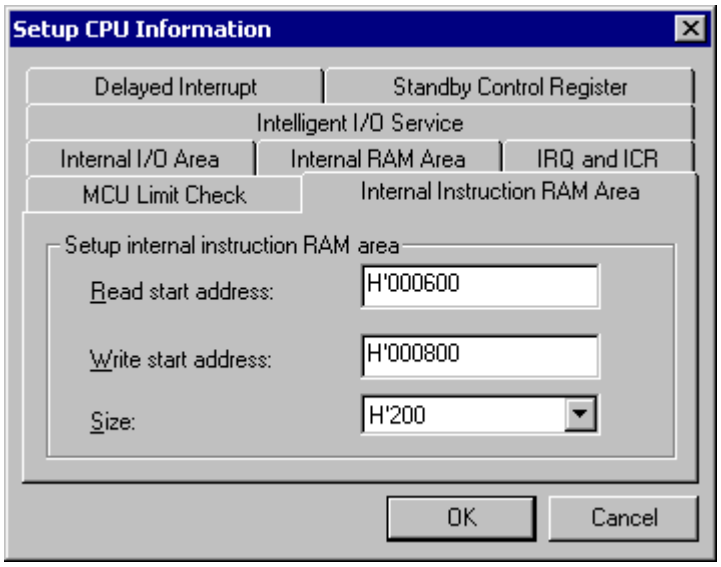


- MCU limit check

  Specifies whether to enable or disable native check (Enable/Disable).

- Setup CPU Information MCU Internal Instruction RAM Area [16H]

Figure 4.5-25 CPU Information Setup Dialog (Internal Instruction RAM Area)



- Read start address

  Specifies the start address of the internal instruction RAM read area.

- Write start address

  Specifies the start address of the internal instruction RAM write area.

- Size

  Specifies the size of the internal instruction RAM area.

Note:

There is no function with some CPU.

4.5.5.3 Setting C Compiler Options

This section explains how to set C compiler options.

■ Setting C compiler options

1. Click the [C Compiler] tab from the project setup dialog.
2. Select category.

   [General], [Define Macro], [Include Path], [Optimize], [Language specifications] or [Target Dependency] can be selected as category.

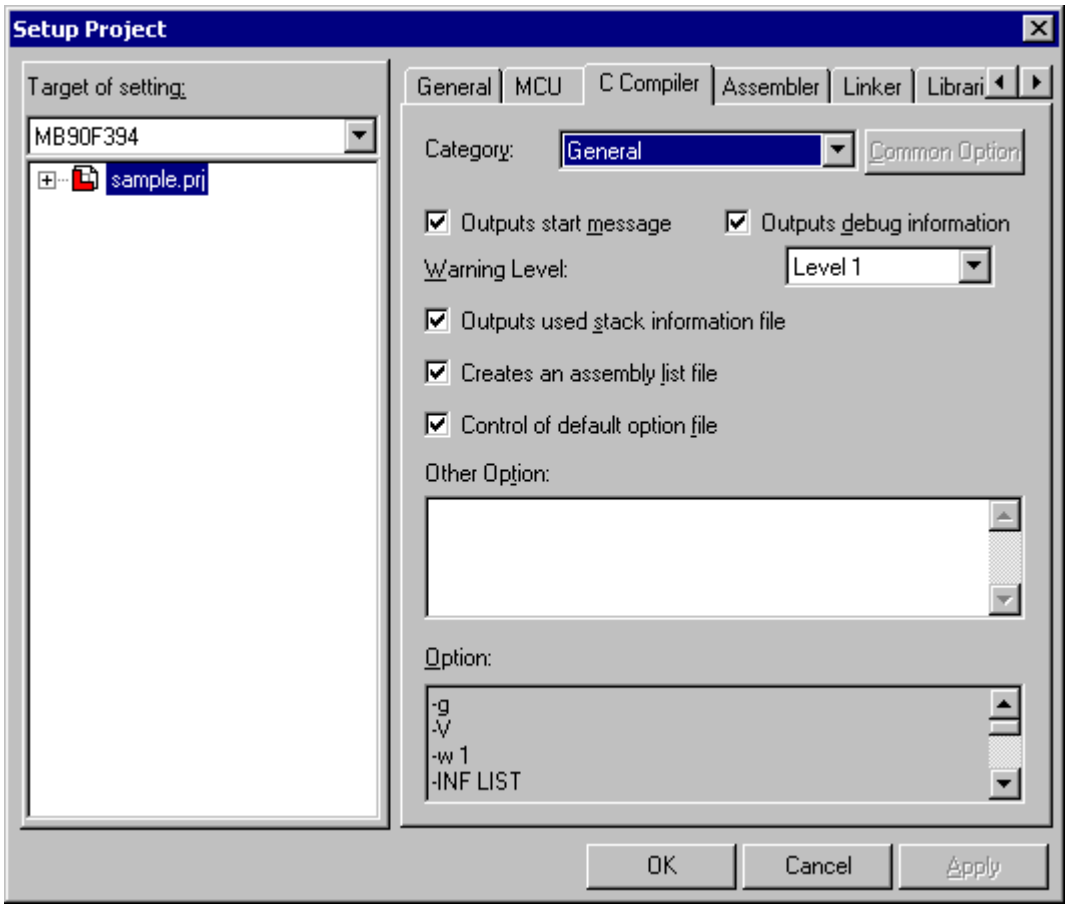
   Specified options can be checked using [Option] at the bottom of the dialog.

   If the individual option is set, the [Common Option] button can be clicked to return to the common option.

   The macro description can be used to describe options. For the macro description, refer to Section "1.11 Macro Descriptions Usable in Manager" of SOFTUNE WORKBENCH User's Manual.

■ Setting [General] Options

Figure 4.5-26 General Option Setup Dialog



The following options can be set from the normal option setup dialog.

- Outputs start message (-V).

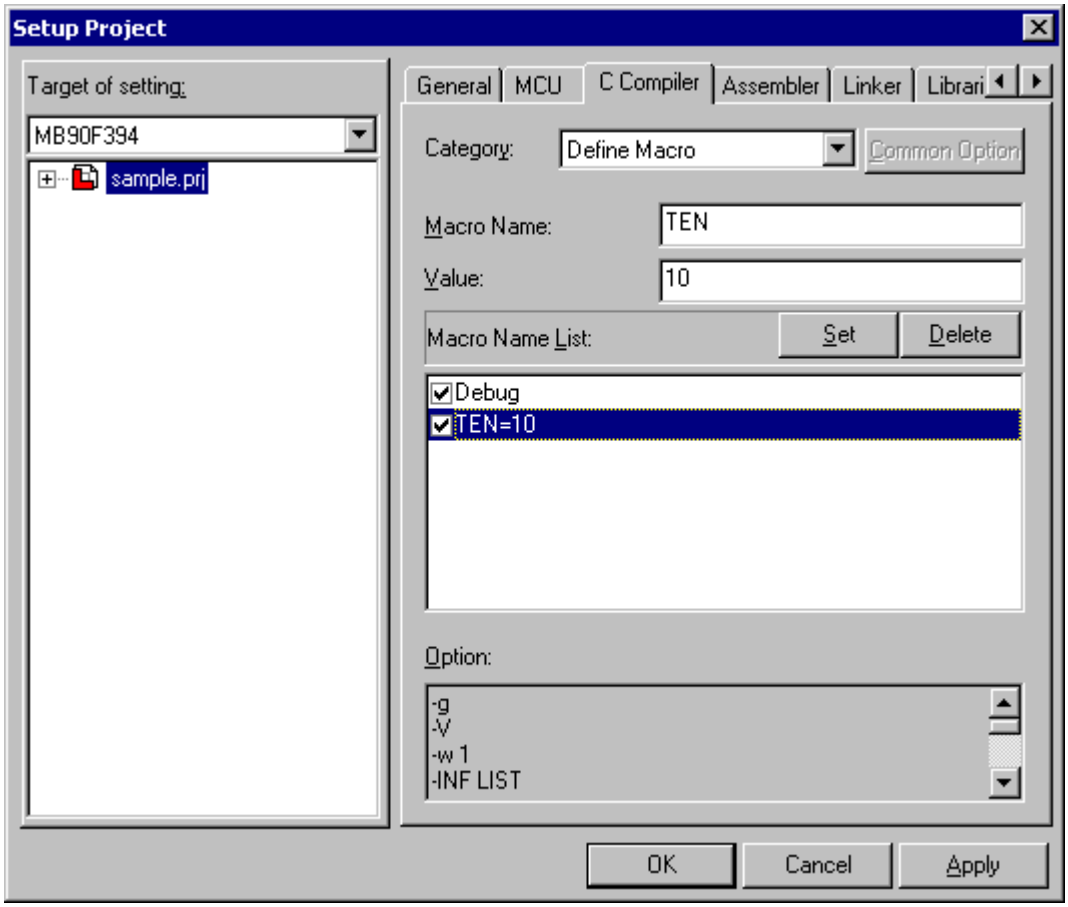
- Outputs debug information (-g).
- Outputs warning message (or warning level) (-w).
- Outputs stack usage information list (-INF STACK).
- Creates assembly list file (-INF LIST).
- Control of default option file (-Xdof).

Depending on CPU classification, [Output warning level] (levels 0 to 3) may be selected instead of [Output warning message].

At Specification in [Other Option], all C compiler options can be written like start-up options from command lines. Write the options that do not belong to any C compiler option setup categories directly at Specification in [Other Option].

■ Setting a macro name

Figure 4.5-27 Macro Name Setup Dialog



If there are two or more items to be set, the macro name found in some item is grayed.

1. Select the [Define Macro] category.
   The macro name setup dialog shown in Figure 4.5-27 opens.
2. Specify the macro name.
3. Specify the setting value as required.
4. Click the [Set] button.
   The specified macro name is set as a define (-D) option.

5. To set the specified macro name as an undefine (-U) option, reset the check mark of the macro name from [Macro Name List].

Note:

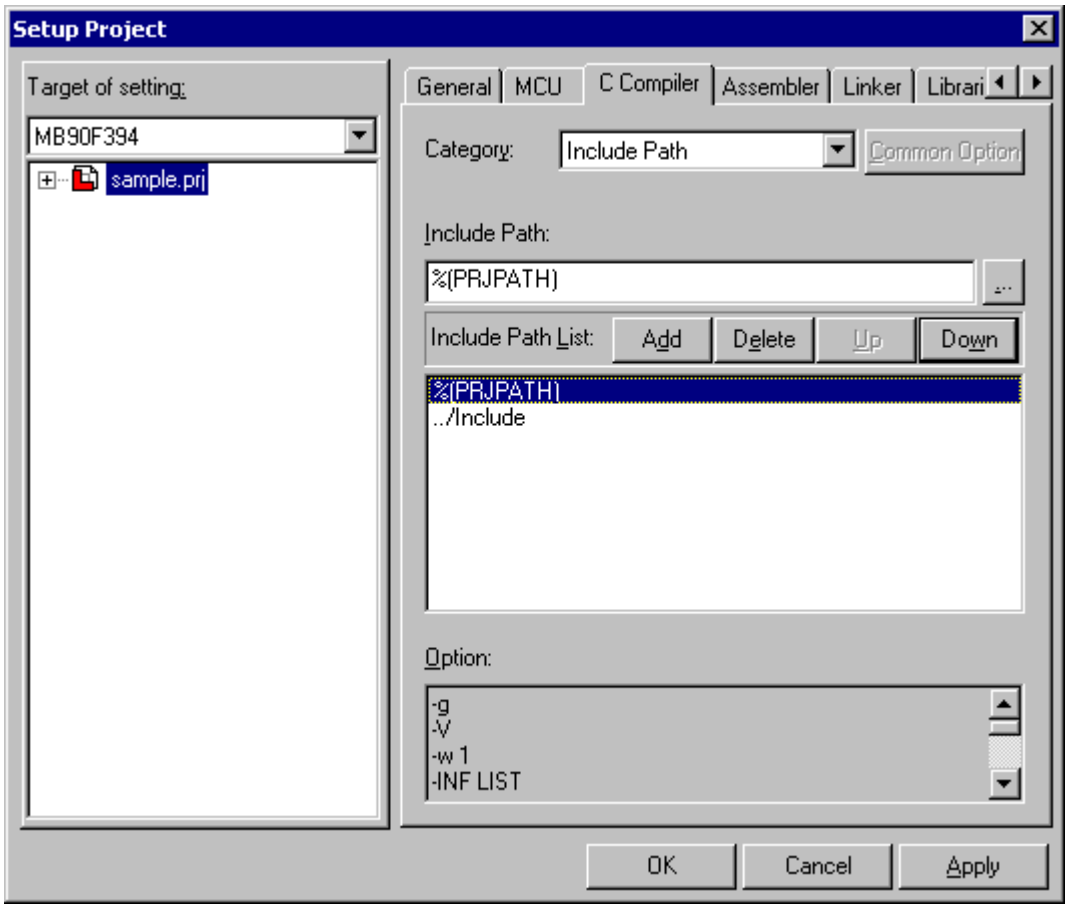
When "undefine" is set, both the define and undefine options are output for the same macro name. This causes no problem because the undefine option precedes the define option.

■ Resetting a macro name

1. Select the [Define Macro] category.
   The macro name setup dialog box shown in Figure 4.5-22 opens.
2. Select the macro name to reset from [Macro Name List].
3. Click the [Delete] button.

■ Setting an include path

Figure 4.5-28 Include Path Setup Dialog



1. Select the [Include Path] category.
   The include path setup dialog shown in Figure 4.5-28 opens.
2. Specify the include path.
   Clicking the [Brows"] button to the right of the input field enables directory selection.

3. Click the [Add] button.
   The specified include path is added to the end of the [Include Path List].

■ Resetting an include path

1. Select the [Include Path] category.
   The include path setup dialog shown in Figure 4.5-28 opens.
2. Select the include path to reset in the [Include Path List].
3. Click the [Delete] button.

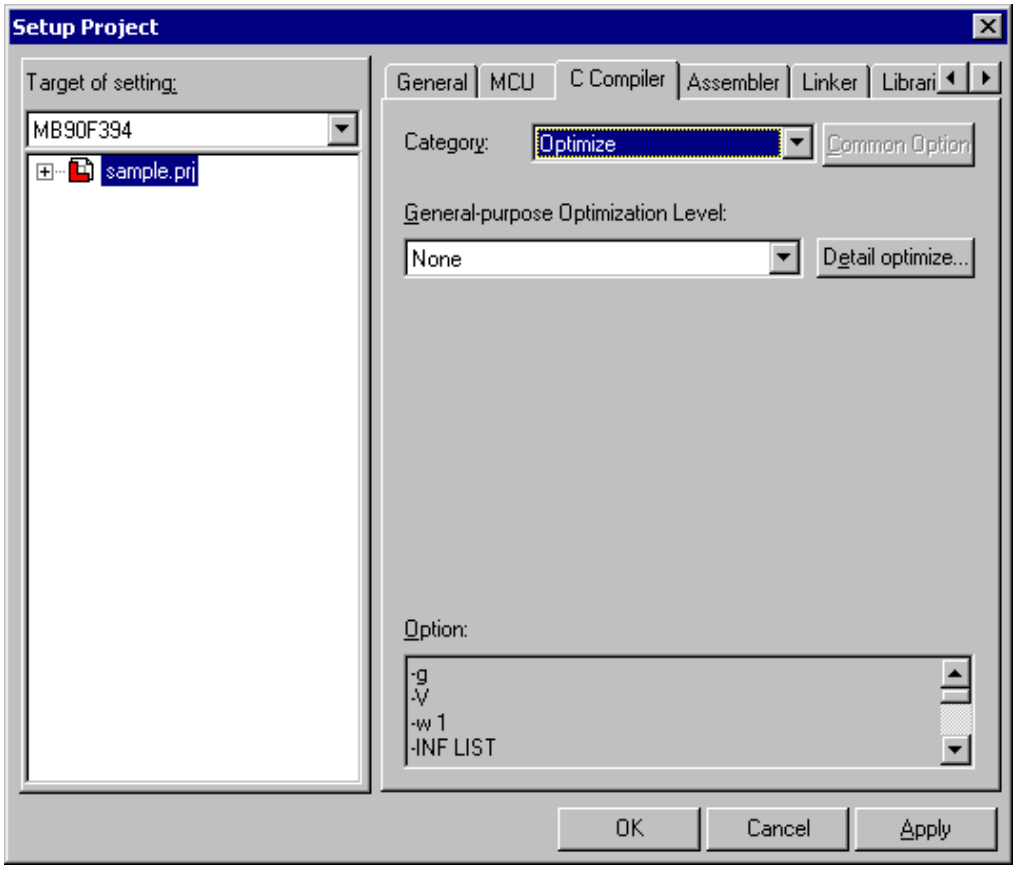
■ Changing the include path retrieval order

Include paths are retrieved in turn from top of the [Include Path List]. The order in which include paths are registered in the list can be changed as follows:

1. Select the [Include Path] category.
   The include path setup dialog shown in Figure 4.5-28 opens.
2. Select the include path whose order is to be changed from [Include Path List].
3. Click the [Up] or [Down] button to move the cursor to a relevant position.

■ Setting [Optimize] options

Figure 4.5-29 Optimize Option Setup Dialog



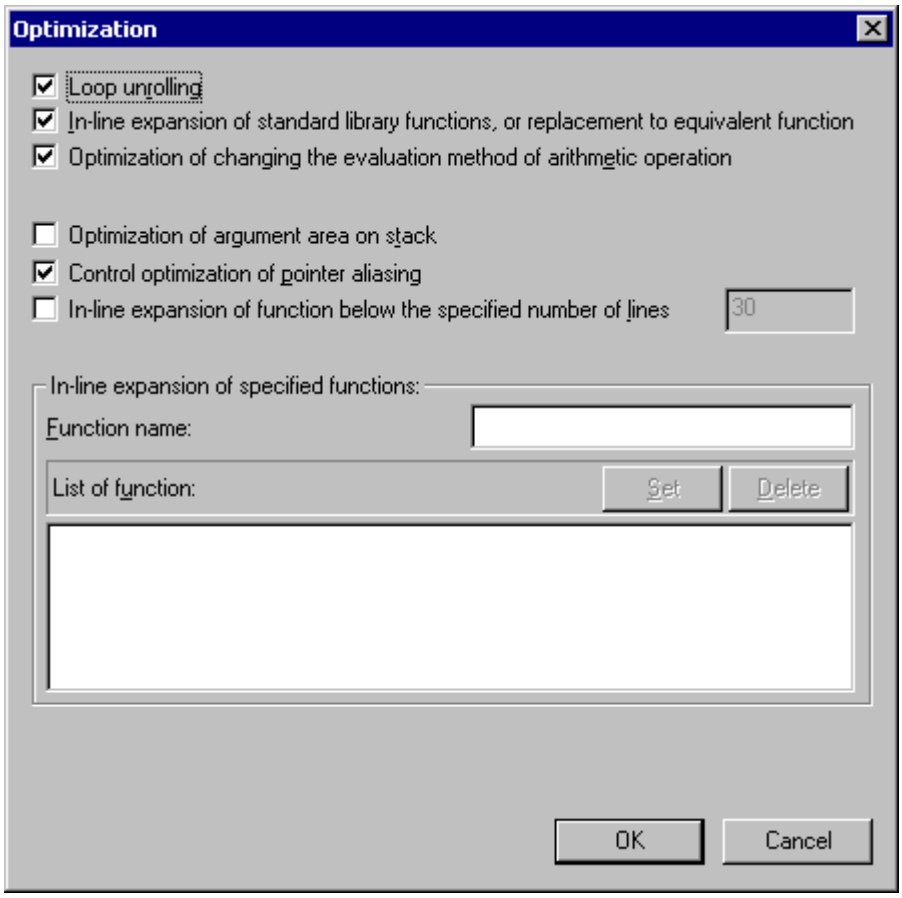
The following options can be set from the optimize option setup dialog. Select the optimize option to set from the drop-down list.

- General-purpose optimization level. (-O)

  (None/Level1-4/speed priority/size priority)

■ Setting of Options Included in [Detail Definition] in [Optimization]

Figure 4.5-30 Optimize Details Setup Dialog



The following options can be specified:

- Loop unrolling (-K UNROLL)
- In-line expansion of standard library functions, or replacement to equivalent function (-K LIB)
- Optimization for changing type of evaluating method of arithmetic operations (-K EOPT)
- Optimization of argument area on stack (-K ADDSP)
- Control optimization of pointer aliasing (-K NOALIAS)
- In-line expansion of function below the specified number of lines (-x auto)
- In-line expansion of specified functions (-x)

■ Setting In-line Expansion of Specified functions

1. Select the [Optimization] category.
   The Set Optimization dialog (See Figure 4.5-30) opened.
2. Set [Optimization level] to any values other than 0 and click [Detail] button.
3. Specify [Function name].
4. Click the [Set] button.

■ Canceling In-line Expansion of Specified functions

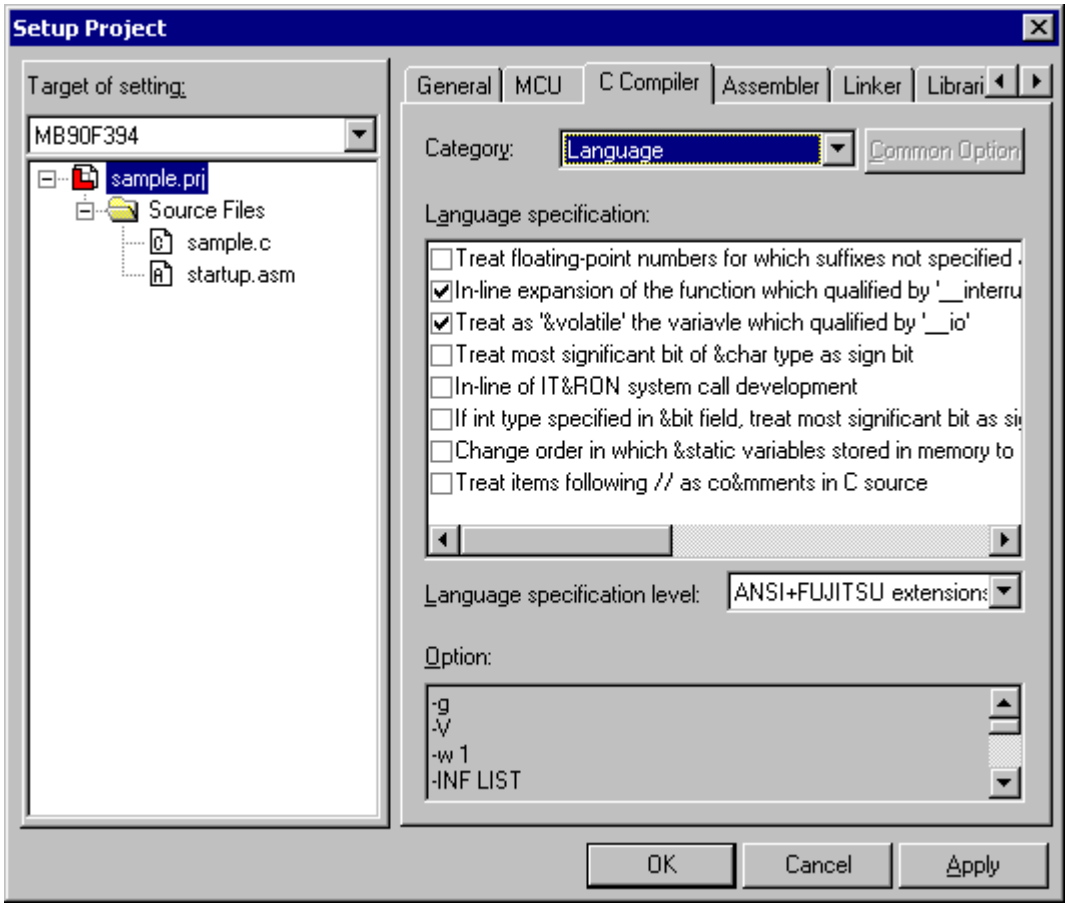
1. Select the [Optimization] category.
   The Set Optimization dialog (See Figure 4.5-30) opened.
2. Sets [Optimization level] to any values other than 0 and click [Detail] button.
3. Select the function name to be deleted from [Function list].
4. Click the [Delete] button.

Note:

If an optimization level is changed, options set in the Detail Definition Dialog are initialized according to the optimization level.

■ Setting of Options Included in [Language Specifications]

Figure 4.5-31 Language Specification Option Setup Dialog



The following options can be specified:

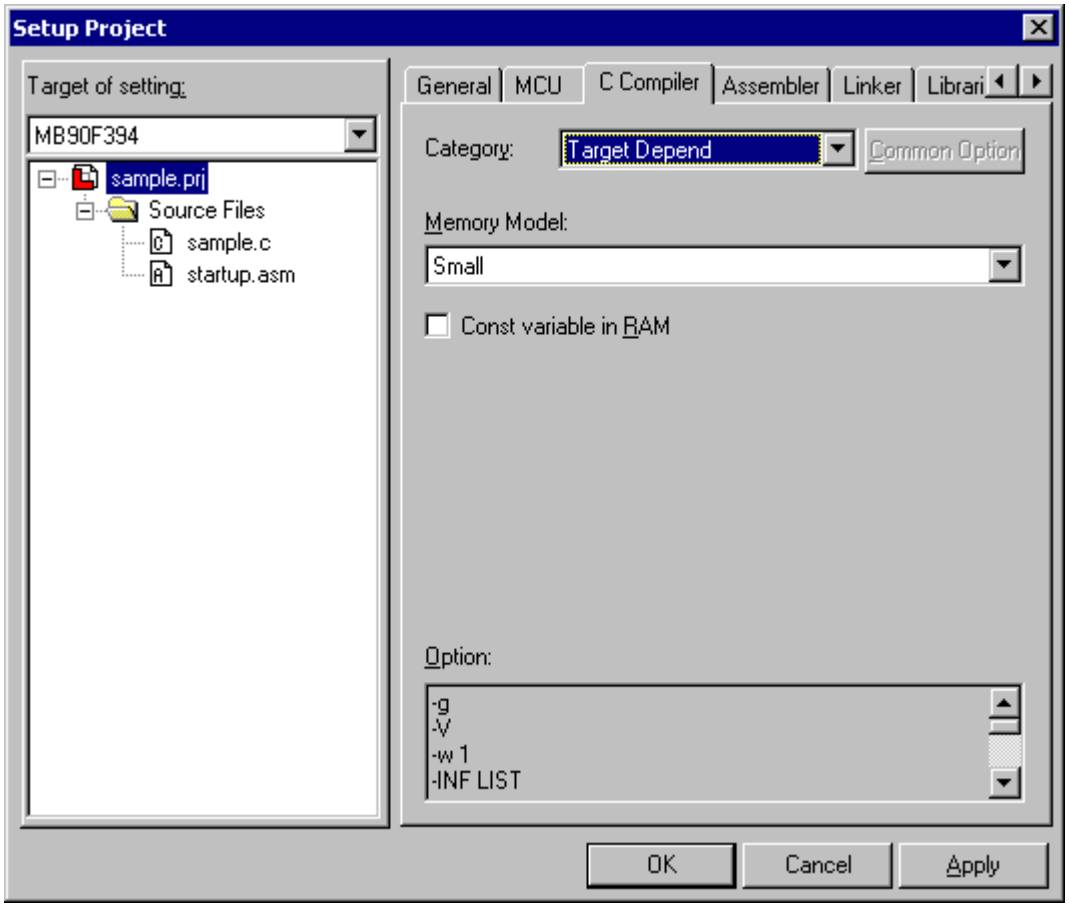
- Treat floating-point numbers for which suffixes are not specified as float type (-K FCONST/DCONST).
- In-line expansion of the function which is qualified by '_interrupt' (-K NOINTLIB).
- Treat the variable which is qualified by '_io' as 'volatile' (-K NOVOLATILE).
- Treat most significant bit of char type as sign bit (-K SCHAR).

- In-line of ITRON system call development (-K REALOS).
- If int type specified in bit field, treat most significant bit as sign bit (-K SBIT).
- Change order in which & static variables are stored in memory to order in which sources are described (-verorder).
- Treat items following " // " as comments in C source (-B).
- Language specification level (-Ja|c|e)

  (ANSI/ANSI + FUJITSU extensions)

■ [Target Depend] options

Figure 4.5-32 Target Dependency Setup Dialog



The following options can be specified:

- Memory Model (-model)
- Const variable in RAM (-ramconst)

4.5.5.4 Setting Assembler Options

This section explains how to set assembler options.

■ Setting assembler options

1. Click the [Assembler] tab from the project setup dialog box.
2. Select category.

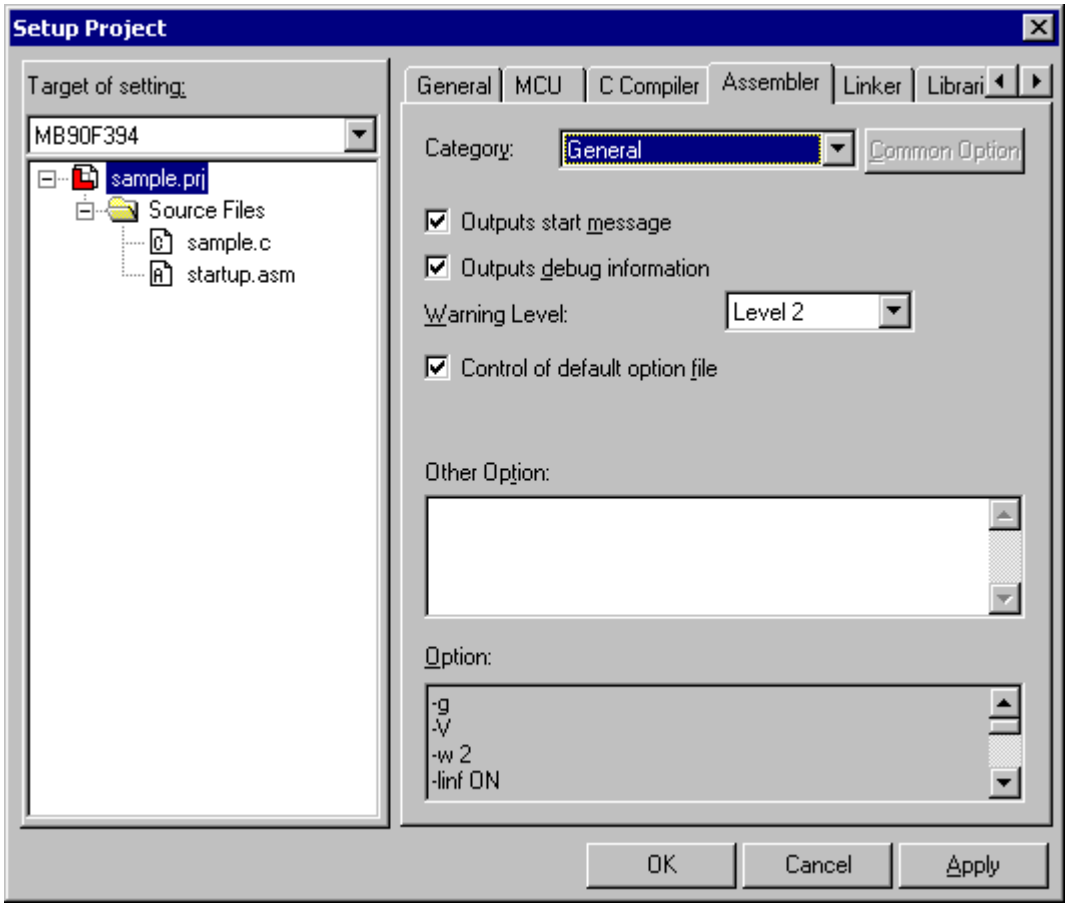
   [General], [Define Macro], [Include Path], [Target Depend], or [Output List] can be selected as category.

   Specified options can be checked using [Option] at the bottom of the dialog.

   If the individual option is set, the [Common Option] button can be clicked to return to the common option. The macro description can be used to describe options. For the macro description, refer to Section "1.11 Macro Descriptions Usable in Manager" of SOFTUNE WORKBENCH User's Manual.

■ Setting [General] Options

Figure 4.5-33 General Option Setup Dialog Box



The following options can be set from the general option setup dialog box.

- Outputs start message (-V)
- Outputs debug information (-g)

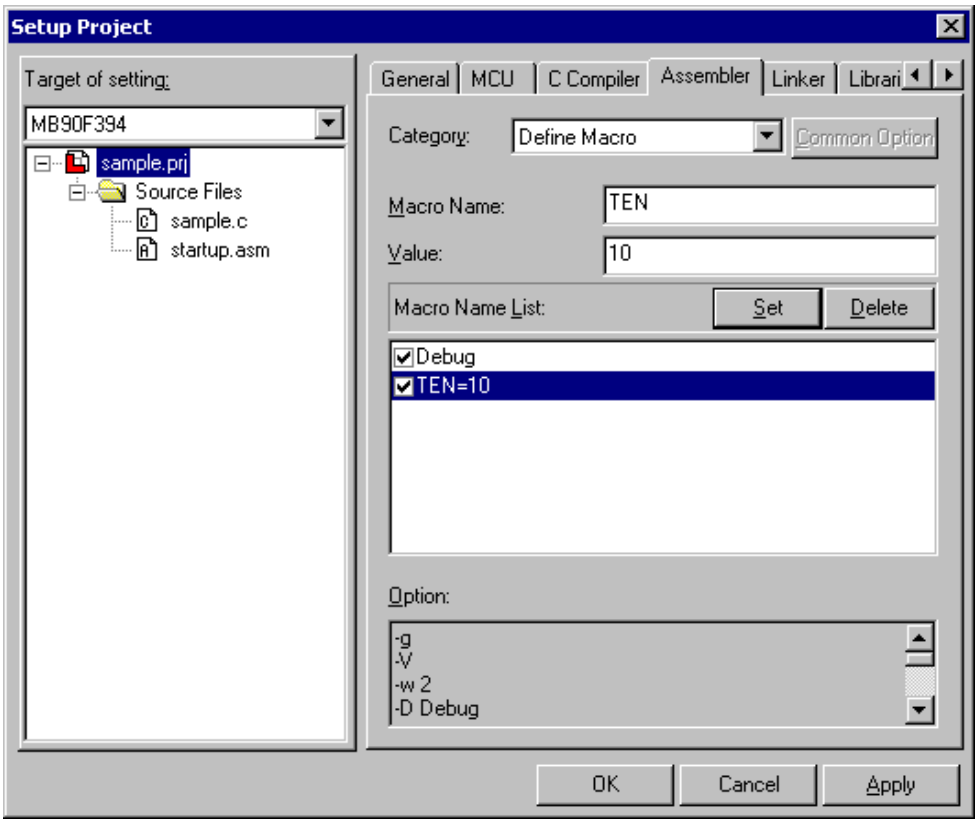
- Control of default option file (-Xdof)
- Warning Level (-w) (Level 0 to 3)

Select Warning Level from the drop-down list.

In [Other Option], all assembler options can be written like startup options from command lines. Write the options that do not belong to any assembler option setup categories directly in [Other Option].

■ Setting a macro name

Figure 4.5-34 Macro Name Setup Dialog Box



If there are two or more items to be set, the symbol found in some item is grayed.

1. Select the [Define Macro] category.
   The macro name setup dialog box shown in Figure 4.5-34 opens.
2. Specify a macro name.
3. Specify a setting value as required.
4. Click the [Set] button.
   The specified macro name is set as a define (-D) option.
5. To set the specified macro name as an undefine (-U) option, reset the check mark of the macro name in [Macro Name List].

Note:

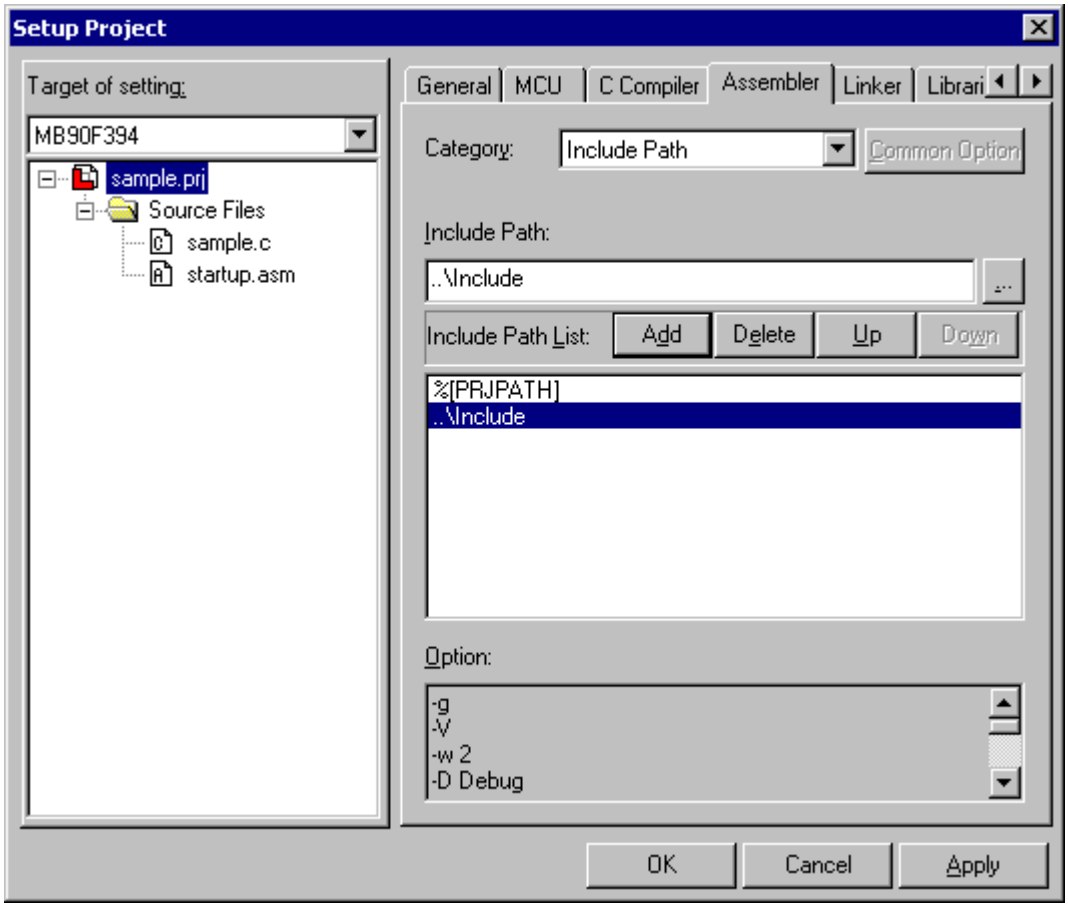
When "undefine" is set, both the define and undefine options are output for the same macro name. This causes no problem because the undefine option precedes the define option.

■ Resetting a macro name

1. Select the [Define Macro] category.
   The macro name setup dialog box shown in Figure 4.5-34 opens.
2. Select the macro name you want to reset from [Macro Name List]
3. Click the [Delete] button.

■ Setting an include path

Figure 4.5-35 Include Path Setup Dialog Box



1. Select the [Include Path] category.
   The include path setup dialog box shown in Figure 4.5-35 opens.
2. Specify an include path.
   Clicking the [Browser button to the right of the input field enables directory selection.
3. Click the [Add] button.
   The specified include path is added to the end of [Include Path List].

■ Resetting an include path

1. Select the [Include Path] category.
   The include path setup dialog box shown in Figure 4.5-35 opens.
2. Select the include path you want to reset from [Include Path List]
3. Click the [Delete] button.

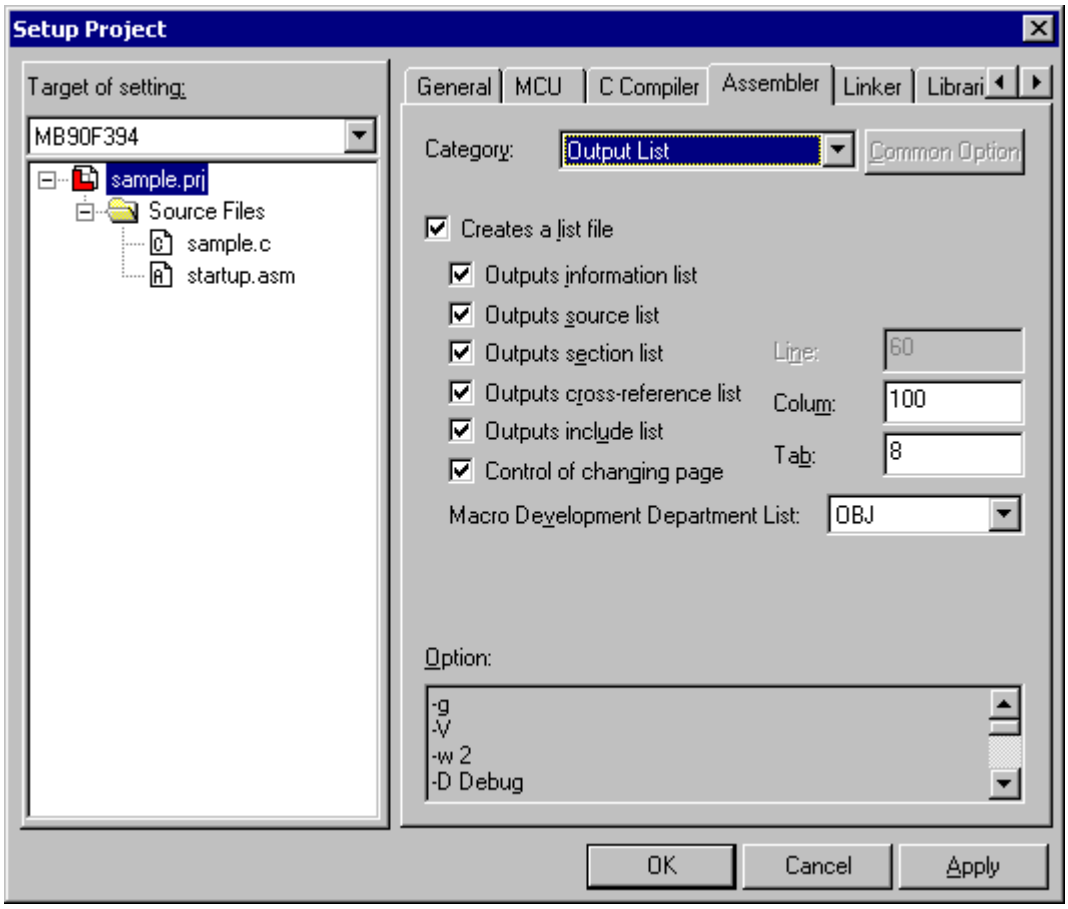
■ Changing the include path retrieval order

Include paths are retrieved in turn from top of [Include Path List]. The order in which include paths are registered in the list can be changed in the following procedure:

1. Select the [Include Path] category.
   The include path setup dialog box shown in Figure 4.5-35 opens.
2. Select the include path whose order is to be changed from [Include Path List].
3. Click the [Up] or [Down] button to move the cursor to a relevant position.

■ Setting list output

Figure 4.5-36 List Output Setup Dialog Box



1. Select the [Output List] category.
   The list output setup dialog box shown in Figure 4.5-35 opens.
2. To output a list file, set a check mark to the left of [Creates a list file].
   When the list file is not output, no other item need to be set. Execute Step (3) and after only when outputting the list file.
3. Select the list file(s) you want to output. Items with check marks are selected.
   - Outputs information list
   - Outputs source list
   - Outputs section list
   - Outputs cross-reference list
   - Outputs include list

4. Specify whether to suppress page change. When a check mark is set to the left of [Control of changing page], page change is suppressed.
   When page change is suppressed, the line count cannot be set.
5. Select a [Macro Development Department List] from the drop-down list. (No output , SRC/OBJ , OBJ)
6. Set line count, column count, and tab count to the right of [Line], [Column], and [Tab] as required.

4.5.5.5 Setting Linker Options

This section explains how to set linker options.

■ Setting linker options

1. Click the [Linker] tab from the project setup dialog box.
2. Select category.

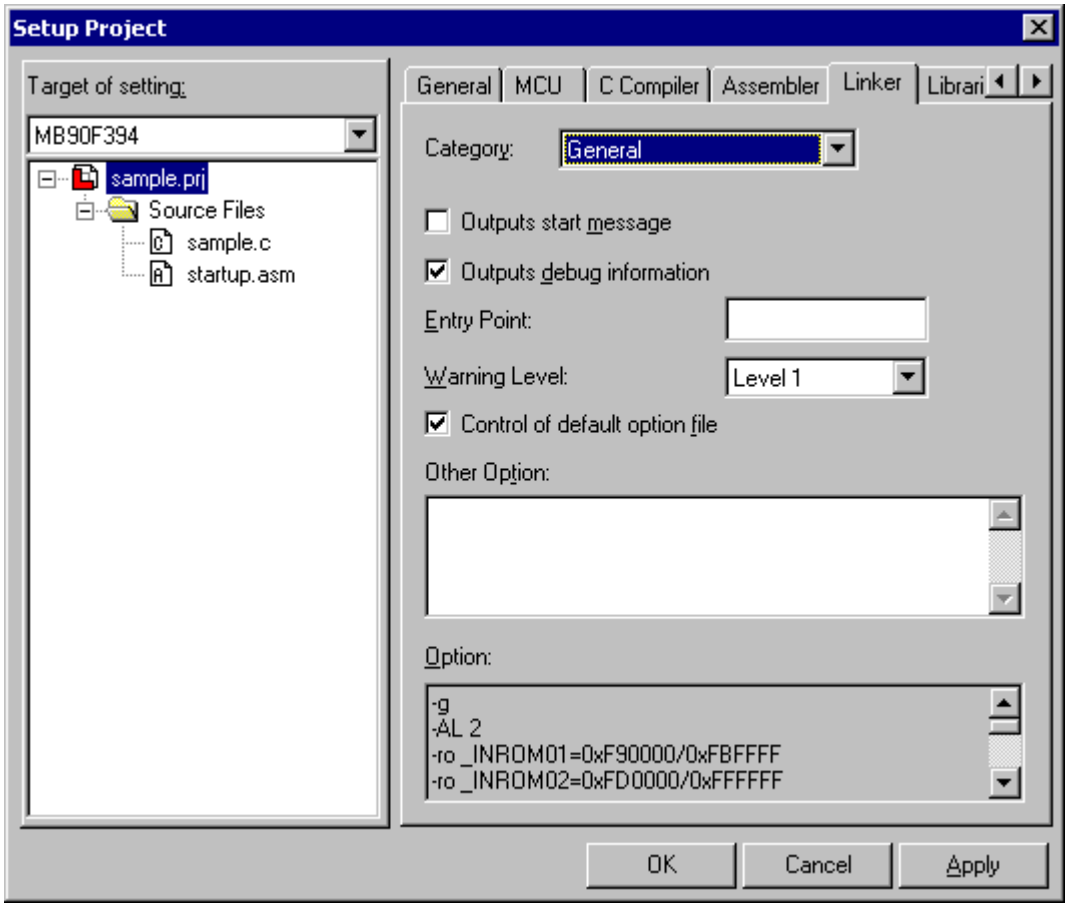
   [General], [Allocation/Link], [Define Symbol], [List Output], [Absolute Assemble List] [Control Library] , or [Register Bank] can be selected as category.

   Specified options can be checked using [Option] at the bottom of the dialog.

   The macro description can be used to describe options. For the macro description, refer to Section "1.11 Macro Descriptions Usable in Manager" of SOFTUNE WORKBENCH User's Manual..

■ Setting [General] options

Figure 4.5-37 General Option Setup Dialog Box



The following options can be set from the general option setup dialog box.

- Outputs start message (-V)
- Outputs debug information (-g)
- Control of Default Option File (-Xdof)

- Entry Point (-e)
- Warning Level (-w)(Level 0 to 2)
- Other Option

Entry Point is the address to be set in the PC when data is loaded by the debugger. Be sure to specify this option with a global symbol.

Select Warning Level from the drop-down list.

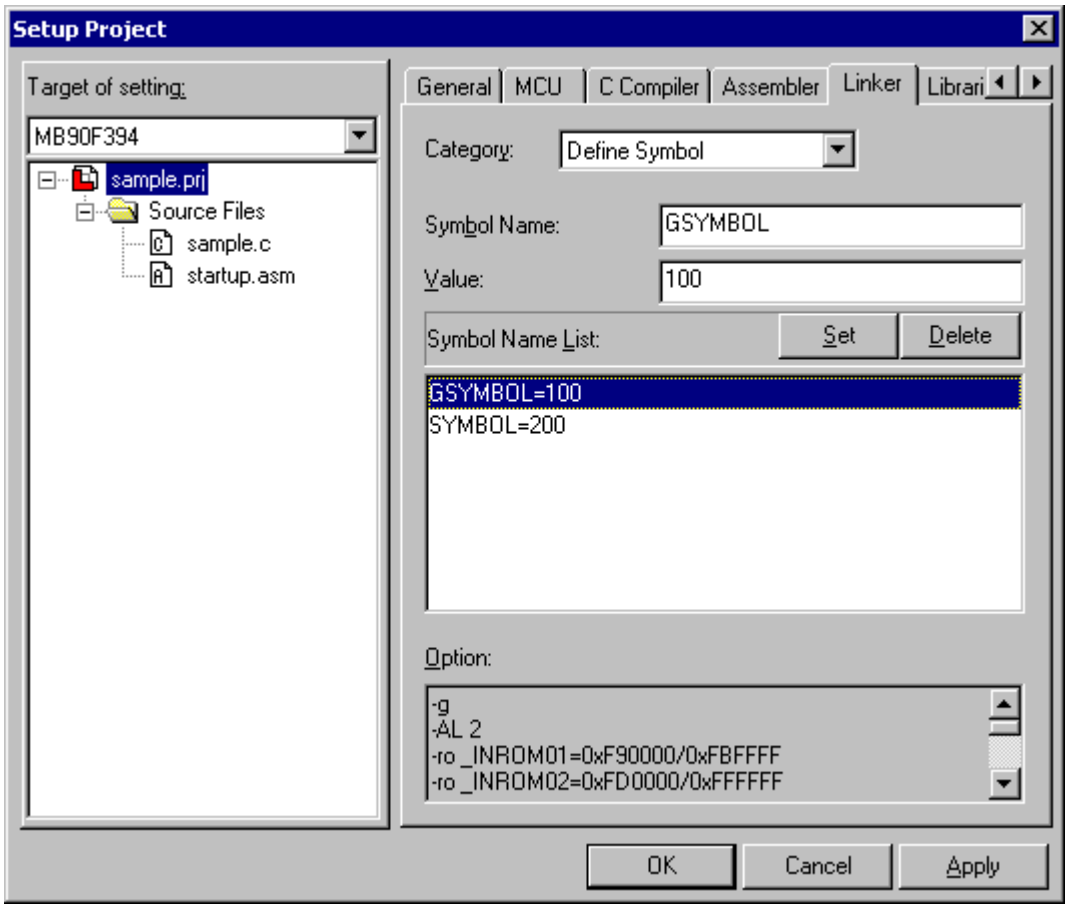
In [Other Option], all assembler options can be written like startup options from command lines. Write the options that do not belong to any assembler option setup categories directly in [Other Option].

■ Setting disposition/connection

See Section "4.5.5.6 Section Disposition/Connection Specifying".

■ Setting symbol definition

Figure 4.5-38 Symbol Definition Setup Dialog Box



If there are two or more items to be set, the symbol found in some item is grayed.

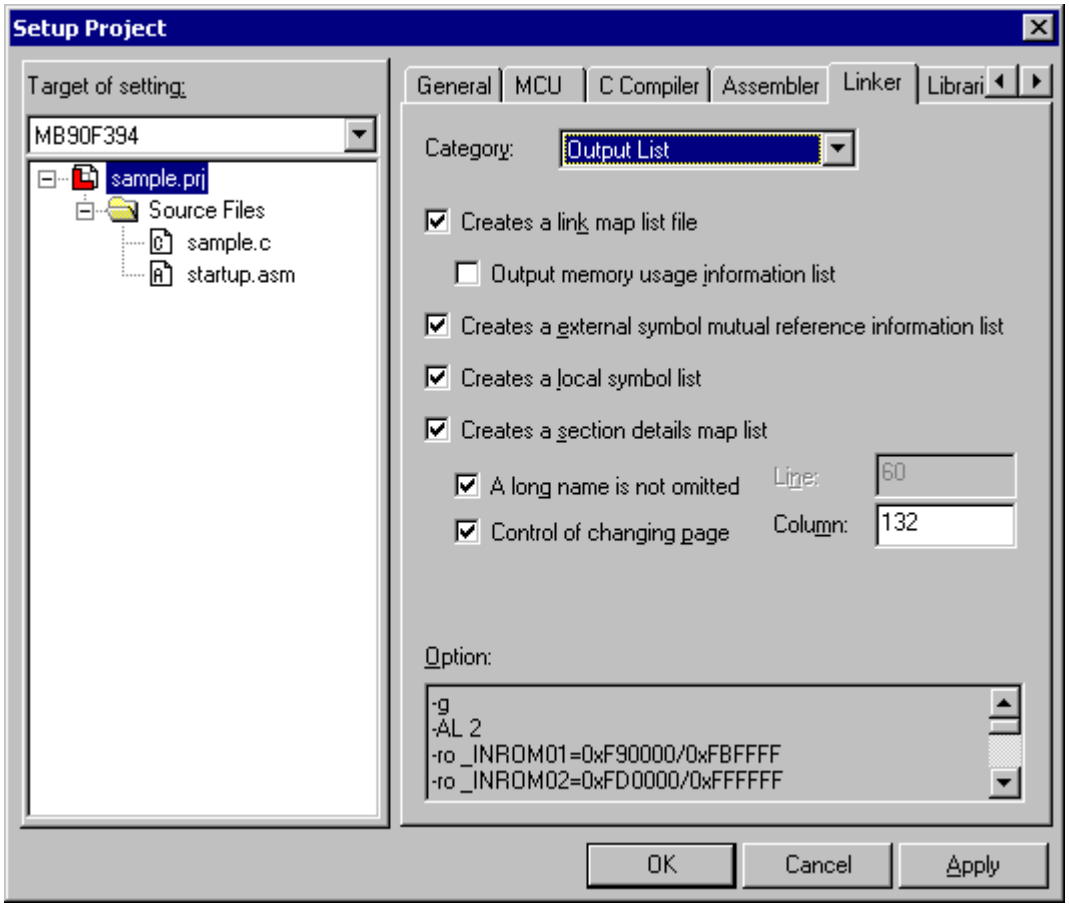
1. Select the [Define Symbol] category
   The symbol definition setup dialog box shown in Figure 4.5-38 opens.
2. Specify a symbol name.
3. Specify a setting value.
4. Click the [Set] button.

■ Resetting symbol definition

1. Select the [Define Symbol] category.
   The symbol definition setup dialog box shown in Figure 4.5-38 opens.
2. Select the symbol name you want to reset from [Symbol Name List].
3. Click the [Delete] button.

■ Selecting output file

Figure 4.5-39 Output List Setup Dialog Box



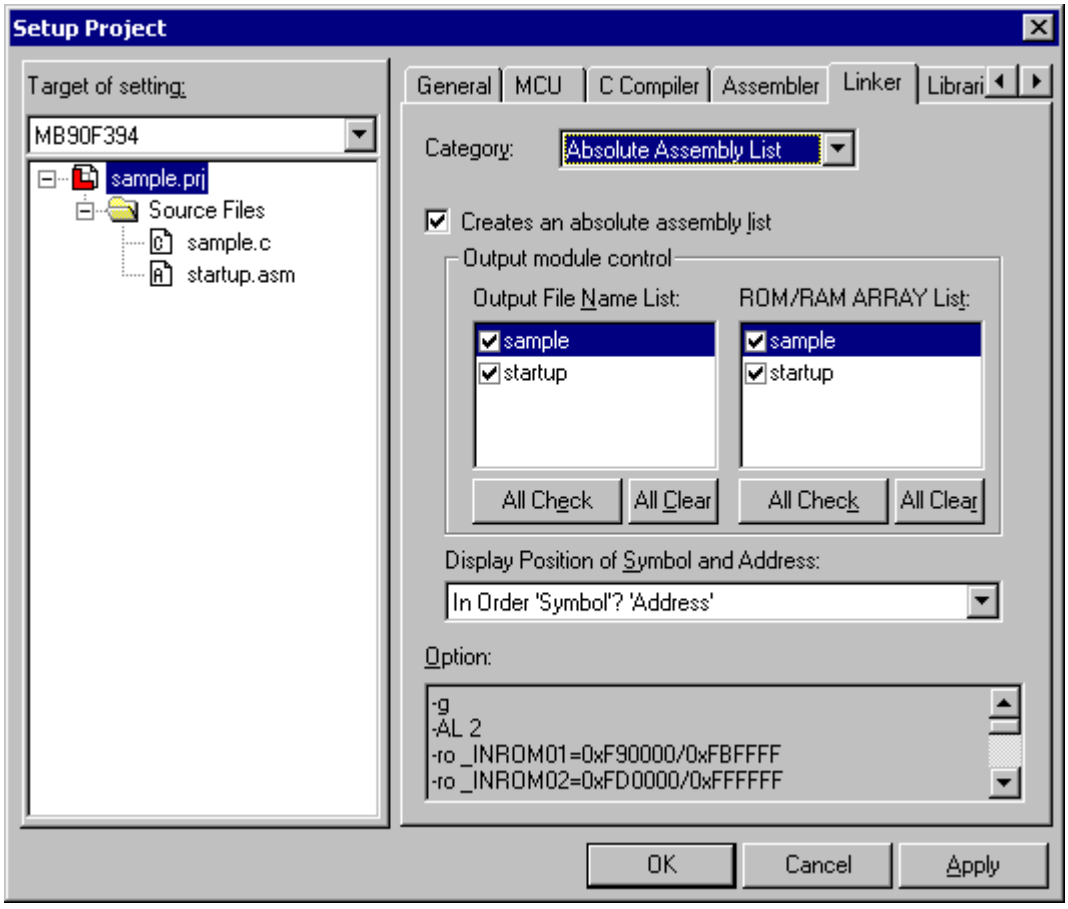
1. Select the [Output List] category.
   The list output setup dialog box shown in Figure 4.5-39 opens.
2. Set check marks to the left of the list(s) to be created.
   - Creates a link map list file
   - Creates a external symbol mutual reference information list
   - Creates a local symbol list
   - Creates a section details map list

   When none of the above lists is output, no other item need to be set from the list output setup dialog box. Execute Step (3) and after only when outputting any of the lists.
3. Check [Output memory usage information list] as needed.
4. Set a check mark to the left of [A long name is not omitted] as required.
   Even names exceeding one list file line are fully output.

5. Specify whether to suppress page change. When a check mark is set to the left of [Control of changing page], page change is suppressed.
   When page change is suppressed, the line count cannot be set.
6. Set line count and column count to the right of [Line] and [Column] as required.

■ Setting absolute format assemble list options

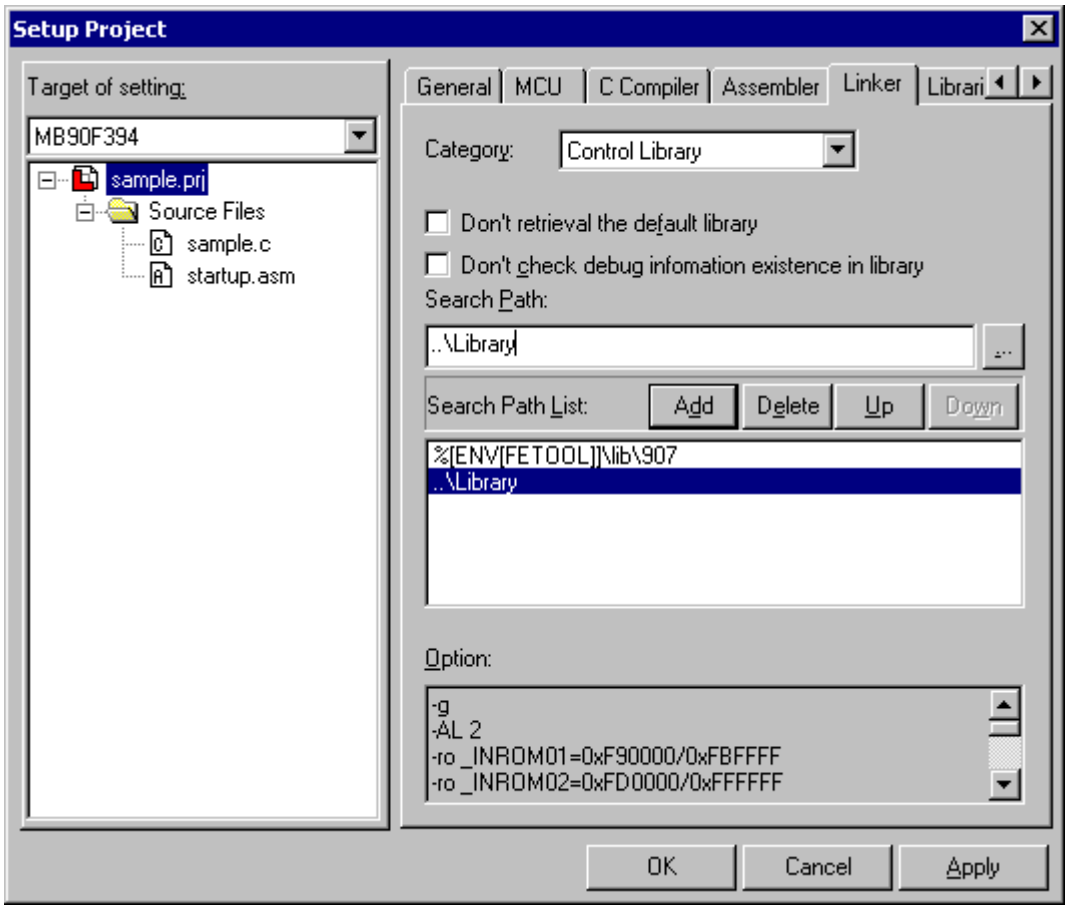
Figure 4.5-40 Absolute Format Assemble List Dialog Box



1. Select the [Absolute Assemble List] category.
   The absolute format assemble list setup dialog box shown in Figure 4.5-40 opens.
2. Set a check mark to the left of [Creates an absolute assemble list].
   When the absolute format assemble list is not output, no other item need to be set from the absolute format assemble list setup dialog box. Execute Step (3) and after only when outputting the absolute format assemble list.
3. Select the module to which the absolute format assemble list is to be output from [Output File Name List], then set a check mark to the left of the selected module.
4. Select the module to which the ROM/RAM ARRAY list is to be output from [ROM/RAM ARRAY List], then set a check mark to the left of the selected module.
5. Select Display Position of Symbol and Address from the drop-down list.
   (In Order 'Address' to 'Symbol' / In Order 'Symbol' to 'Address')

■ Setting of Options for Library Control

Figure 4.5-41 Library Control Setup Dialog Box



■ Setting when the default library is not used

1. Select the [Control Library] category.
   The library control setup dialog box shown in Figure 4.5-41 opens.
2. Set a check mark to the left of [Don't retrieve the default library].

■ Setting method when debug information existence check for library file module is inhibited

1. Select the [Control Library] category.
   The library control setup dialog box shown in Figure 4.5-41 opens.
2. Set a check mark to the left of [Don't check debug information existence in library].

■ Setting a library retrieval path

1. Select the [Control Library] category.
   The library control setup dialog box shown in Figure 4.5-41 opens.
2. Set [Search Path].
   Clicking the [Browse] button to the right of the input file enables path reference.
3. Click the [Add] button.
   The set library retrieval path is added to the end of [Search Path List].

■ Resetting a library retrieval path

1. Select the [Control Library] category.
   The library control setup dialog box shown in Figure 4.5-41 opens.
2. Select the path you want to reset from [Search Path List].
3. Click the [Delete] button.

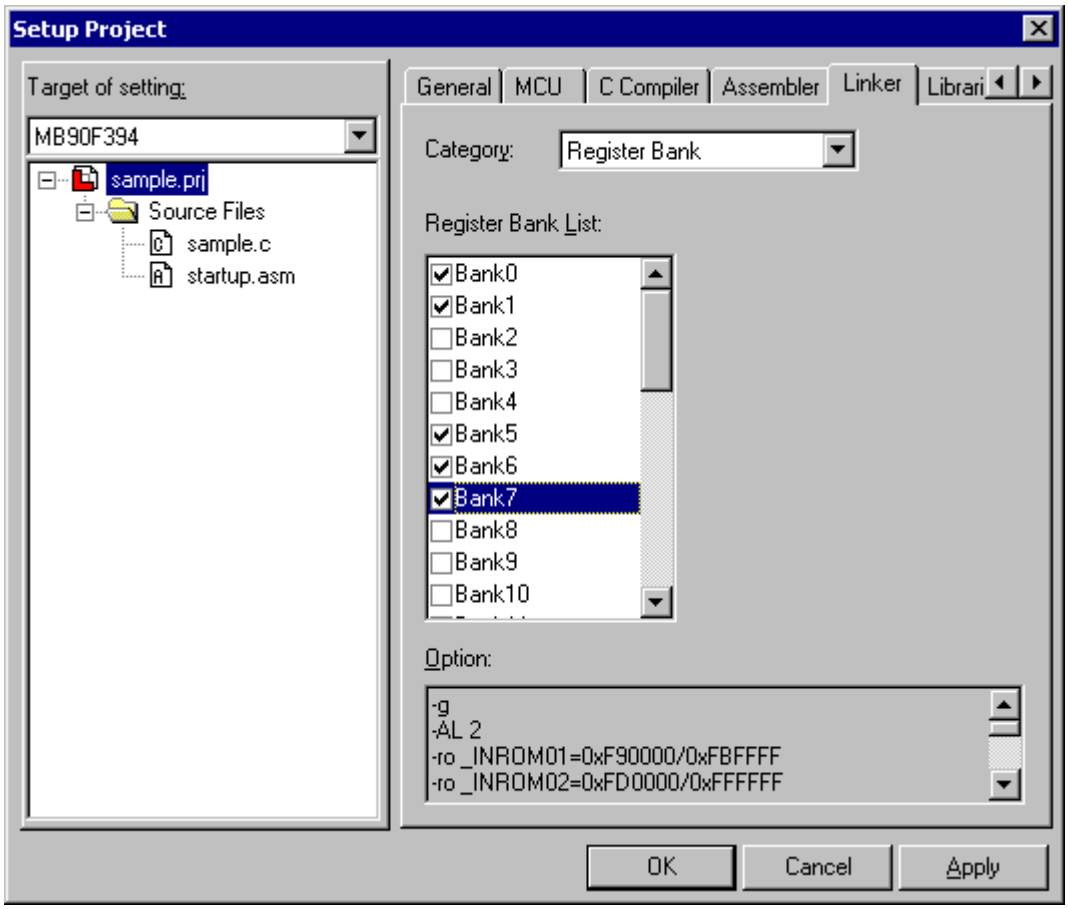
■ Changing the library retrieval path retrieval order

Library retrieval paths are retrieved in turn from top of [Search Path List]. The order in which library retrieval paths are registered in the list can be changed in the following procedure:

1. Select the [Control Library] category.
   The library control setup dialog box shown in Figure 4.5-41 opens.
2. Select the library retrieval path whose order is to be changed from [Search Path List].
3. Click the [Up] or [Down] button to move the cursor to a relevant position.

■ Setting register bank

Figure 4.5-42 Register Bank Setup Dialog Box



1. Select the [Register Bank] category.
   The register bank setting dialog is displayed (Figure 4.5-42).
2. Set the check mark for the register bank list to be used.

4.5.5.6 Section Disposition/Connection Specifying

This section explains how to set the section allocation/link option (linker option).

■ Section disposition/connection specifying

1. Click the [Linker] tab from the project setup dialog box.
2. Select the [Disposition/Connection] category.
   The disposition/connection setup dialog box shown in Figure 4.5-43 opens.
3. Select [Auto Disposition] from the drop-down list (None/Mode1/Mode2).

   - NONE

     The linker does not allocate the sections automatically.

   - Mode1

     If any absolute sections exist when allocating sections in a ROM/RAM area specified in the [ROM/RAM Area List], the linker allocates relocatable sections to avoid an overlap to each of the addresses to be allocated. It will properly allocate the sections in a descending order of their alignment values and sizes so that empty area is minimized.

   - Mode2

     The linker determines whether to allocate sections unspecified for allocation in a ROM area or in a RAM area based on the types of the sections and automatically allocates them in empty area in each area.

Figure 4.5-43 Disposition/Connection Setup Dialog Box

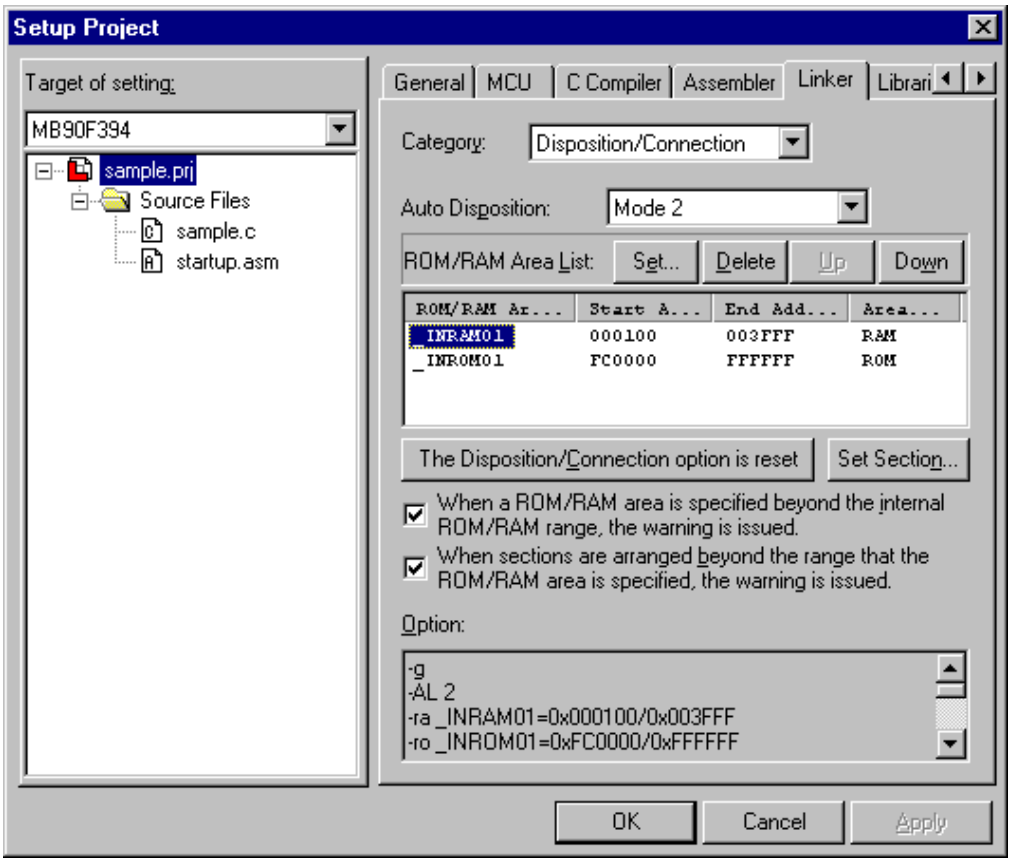
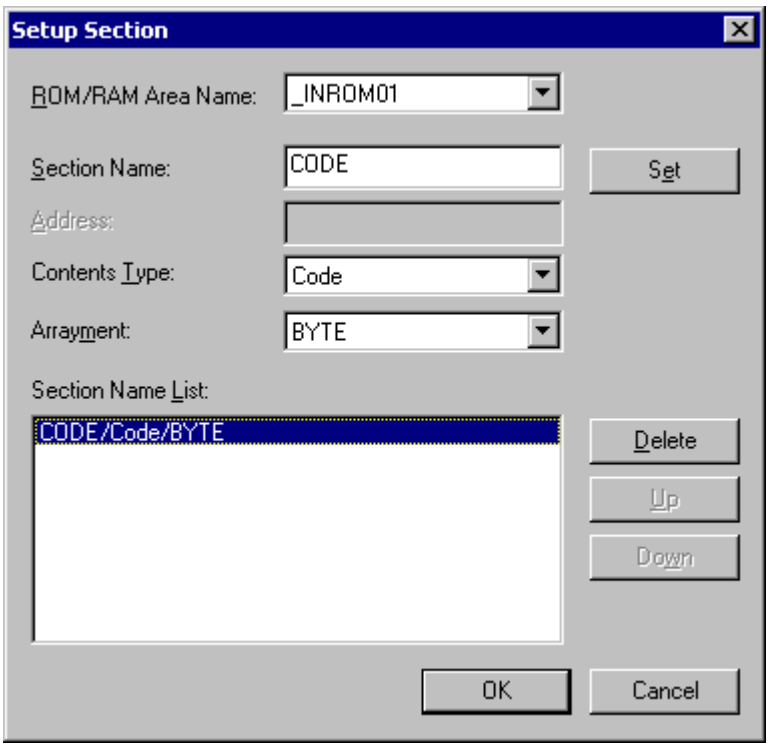


Figure 4.5-44 Setup Section Dialog



■ When a section is allocated to the specified ROM/RAM area

1. Define the ROM/RAM area.
   See Defining the ROM/RAM area.
2. Select the ROM/RAM area you want to set from [ROM/RAM Area List].
3. Click the [Set Section...] button.
   The section setup dialog box shown in Figure 4.5-45 opens.
4. Specify the name of the section to be allocated to the selected area.
   A wild card character can be also used.
5. Select a content type from the drop-down list as required.
   (None/Code/Data/Stack/Const/IO)
6. Click the [Set] button.
   The set section is added to the end of [Section Name List].
7. When all settings are completed, click the [OK] button.
   The order in which sections are allocated to the specified ROM/RAM area is the same as the order in [Section Name List]. For how to change this order, see Changing the section allocation order next by one.

■ When a section is allocated to the specified starting address

1. Click the [Set Section ...] button.
   The section setup dialog box shown in Figure 4.5-45 opens.
2. Select [Specify in Address] from the drop-down list for [ROM/RAM Area Name].
3. Specify the name of the section to be allocated to the ROM/RAM area.
   A wild card can be also used.
4. Specify the starting address to which the section is to be allocated.
5. Select and set a content type from the drop-down list as required.

6. Click the [Set] button.
   The specified section is added to the end of [Section Name List].
7. When all settings are completed, click the [OK] button.
   To continuously allocate several sections to the specified address, execute step (6), set Section Name and Content Type only, then click the [Set] button. Repeat this operation.
   Sections are allocated in the specified order. For how to change this order, see the next changing the section allocation order.

■ Changing the section allocation order

1. Click the [Set Section...] button.
   The section setup dialog box shown in Figure 4.5-45 opens.
2. Specify the area you want to change in [ROM/RAM Area Name]. When an address is directly specified, select [Addressing]. The sections to be allocated to the area (or address) are displayed in [Section Name List].
3. Select the section name whose order is to be changed, then click the [Up] or [Down] button to move the cursor to a relevant position.
   For addressing, also see Notes in addressing in the next item.

■ Notes in addressing

Examples of relationship between descriptions in [Section Name List] and the linker options to be generated are given below. Pay special attention when changing the section allocation order.

[Example 1] When descriptions in [Section Name List] are as follows:

sec1=H'00001000

sec2

sec3=H'0000F000

sec4

The following sections are generated.

-sc sec1+sec2=H'00001000

-sc sec3+sec4=H'0000F000

[Example 2] When the sec4 allocation order is changed as follows:

sec1=H'00001000

sec2

sec4

sec3=H'0000F000

The following sections are generated:

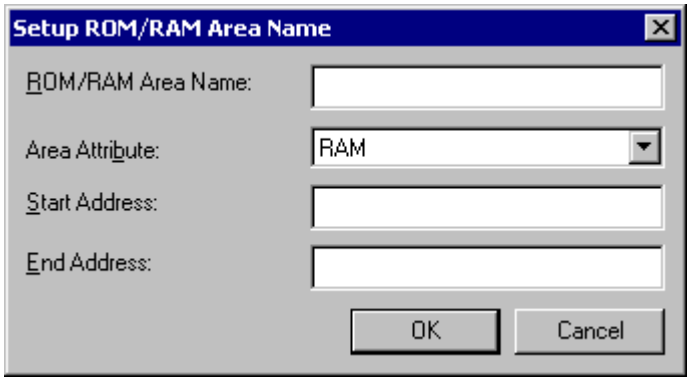
- sc sec1+sec2+sec4=H'00001000

-sc sec3 =H'0000F000

■ Defining the ROM/RAM area

1. Click the [Add] button.
   The "Setup ROM/RAM Area Name dialog (Figure 4.5-45)" is opened.

Figure 4.5-45 Setup Disposition / Connection Dialog



2. Specify a ROM/RAM area name.
   Specify the ROM/RAM name that differs from the set names.
3. Select ROM or RAM as [Area Attribute].
4. Specify a start address and end address (starting address and end address of ROM/RAM area).

In Auto Disposition (Mode 2), the linker searches an available ROM/RAM area beginning at the top of the [ROM/RAM Area List]. Click the [Up] button or the [Down] button to move to a proper position.

■ Deleting a ROM/RAM area

1. Select the ROM/RAM area you want to delete from [ROM/RAM Area List].
2. Click the [Delete] button.

■ Method of outputting warning when ROM/RAM area is specified outside range of internal ROM/RAM

1. Put a check mark in the checkbox. When sections are arranged beyond the range of specified ROM/RAM area, the warning is issued.
   - When an area outside the internal-ROM/RAM area is specified in the dialog, a warning dialog is displayed.
   - When an area outside the internal-ROM/RAM area is set, a warning is output at linking.

■ Method of outputting warning when section is placed outside specified ROM/RAM area range

1. Put a check mark in the checkbox. When a ROM/RAM area is specified beyond the internal ROM/RAM range, the warning is issued.
   - When a section is placed outside the specified ROM/RAM area (-ro and -ra options) or outside the internal-ROM/RAM area in the MCU, a warning is output at linking.

■ When initializing [Placement/link] option based on MCU information

1. Click the [Reset placement/link option] button.

   - The placement/link option is reset.

When this reset is performed, the placement/link option is set as follows:

- Automatic placement (-AL 2)

  Mode 2 (optimum automatic placement by linker) is set.

- ROM/RAM area (-ro, -ra)

  The ROM/RAM area is all cleared and the currently selected MCU internal ROM/RAM address is set.

- Section information (-sc)

  Section information is all cleared.

- A warning is output when a ROM/RAM area outside the range of the internal ROM/RAM (-check_rora) is specified.

  Setting is performed so as to output a warning. However, when an MCU without single-chip mode is selected, setting is performed so as not to output a warning.

- A warning is output when a section is placed outside the specified ROM/RAM area range (-check_locate)

  Setting is performed so as to output a warning.

4.5.5.7 Setting Librarian Options

This section explains how to set librarian options.

■ Setting librarian options

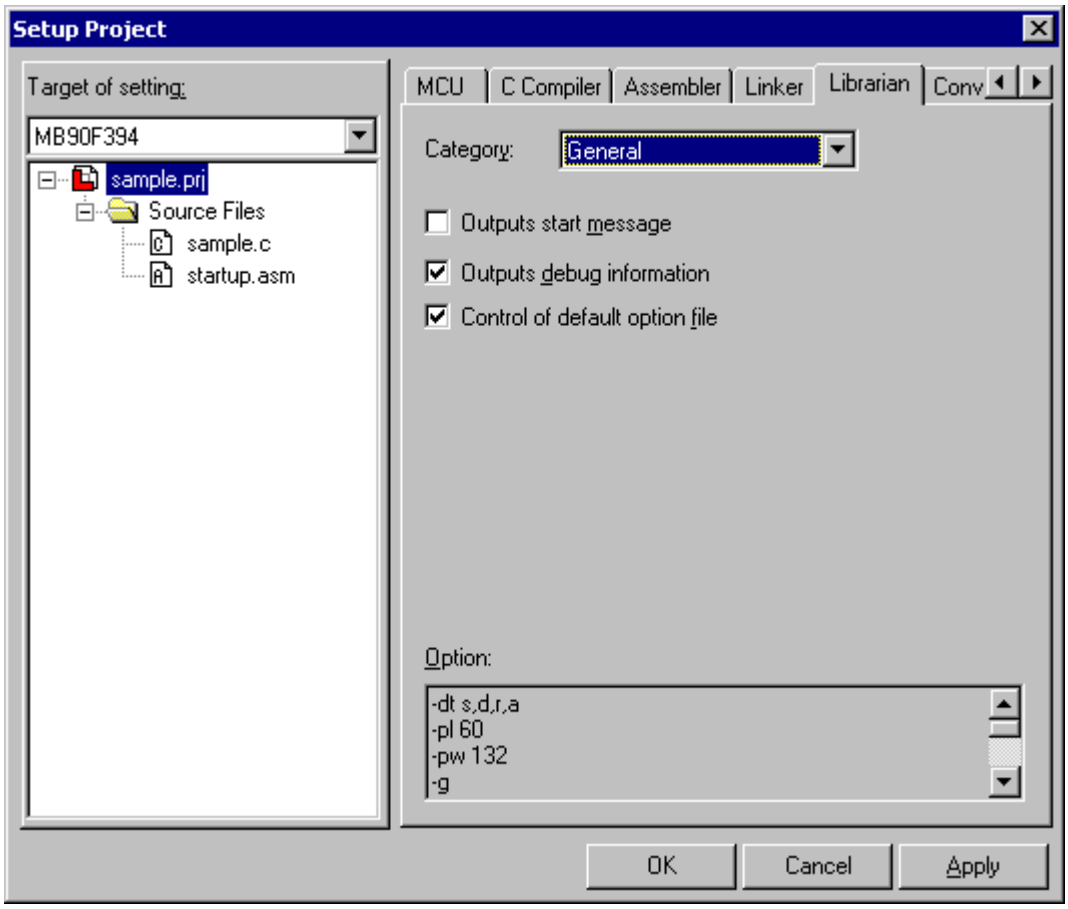
1. Click [Librarian] from the project setup dialog box.
2. Select category.

   [General] or [Output List] can be selected as category.

   The specified options are displayed in the lower part of the project setup dialog box.

■ Setting Options in [General]

Figure 4.5-46 Librarian Option Setup Dialog Box (General)

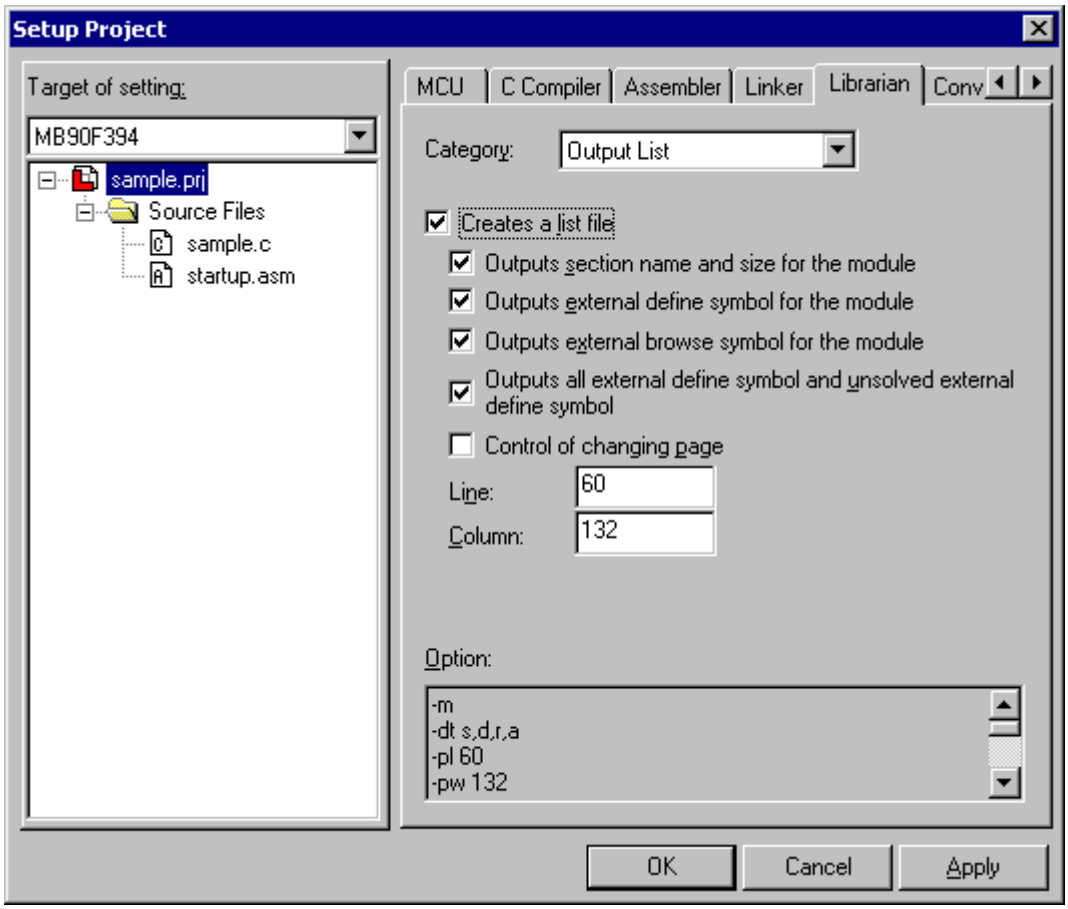


The following options can be set:

- Output start message (-v)
- Output debug information (-g)
- Disable default option (-Xdof)

■ Setting the Output-listing

Figure 4.5-47 Librarian Option Setup Dialog Box (Output List)



1. Select any of the following output types:
   - Outputs section name and size for module
   - Outputs external define symbol for module
   - Outputs external browse symbol for module
   - Outputs all external define symbol and unsolved external define symbol
2. To suppress page change, set a check mark to the left of [Control of changing page]. When page change is suppressed, the line count cannot be set.
3. Set Line Count and Column Count to the right of [Line] and [Column] as required.

■ Starting Librarian

If the project type is a "library file," the librarian is started.

To change the project type, see "4.5.5.1 General".

4.5.5.8 Setting Converter Options

This section explains how to set converter options.

■ Setting converter options

1. Click the [Converter] tab from the project setup dialog box.
2. To start the load module converter, mark the [Start load module converter] checkbox.
3. Select a conversion format.
   - Motorola S format(f2ms)
     - Converts the absolute format load module of linker output to an S format.

       Data at addresses 0 to 0xFFFFFFFF is converted.
   - 32-bit Intel HEX format(f2hs)
     - Converts the absolute format load module of linker output to an HEX format. Data at addresses 0 to 0xFFFFFFFF is converted.
   - Intel HEX format(f2is)
     - Converts the absolute format load module of linker output to an HEX8 format. Data at addresses 0 to 0xFFFF is converted.
     - This format is left to maintain compatibility with the previous version.

       The 32-bit Intel HEX format (f2hs) should be used for conversion to an HEX8 format.
   - Intel Extend HEX format(f2es)
     - Converts the absolute format load module of linker output to an HEX16 format. Data at addresses 0 to 0xFFFFF is converted.
     - This format is left to maintain compatibility with the previous version.

       The 32-bit Intel HEX format (f2hs) should be used for conversion to an HEX16 format.
4. Set the following items as required:
   - Common Options
     - Outputs start message (-V)
     - Control of default option file (-Xdof)
   - Options for Motorola S format (f2ms)
     - Output file format - None

       Outputs data in mixed-S1, S2, and S3 records according to the data address.
     - Output file format - S1 record (16-bit address) (-S1)

       Outputs data in an S1 record (in an allowable output range of 0x0000 to 0xFFFF).
     - Output file format - S2 record (24-bit address) (-S2)

       Outputs data in an S2 record (in an allowable output range of 0x000000 to 0xFFFFFF).
     - Output file format - S3 record (32-bit address) (-S3)

       Outputs data in an S3 record (in an allowable output range of 0x00000000 to 0xFFFFFFFF).
     - Adjust (-adjust)

       Automatically calls the Format Adjuster to adjust a data output format.

- Output range (-ran)

  Specifies the range to be adjusted by an address when selecting the option (-adjust) for adjusting an output file.

  Selecting [Auto] will obtain the starting/ending address for adjustment from the absolute format load module to set automatically.

- padding data (-p)

  When selecting the option (-adjust) for adjusting an output file, the area of the file where no data exists is packed with data having a specified value.

• Options for 32-bit Intel HEX format (f2hs)

- Output file format - None

  Outputs data in mixed-HEX8, HEX16, and HEX32 according to the data address.

- Output file format - HEX8 (16-bit address) (-I16)

  Outputs data in HEX8 format (in an allowable output range of 0x0000 to 0xFFFF).

- Output file format - HEX16 (20-bit address) (-I20)

  Outputs data in HEX16 format (in an allowable output range of 0x00000 to 0xFFFFF).

- Output file format - HEX32 (32-bit address) (-I32)

  Outputs data in HEX32 format (in an allowable output range of 0x00000000 to 0xFFFFFFFF).

- Adjust (-adjust)

  Automatically calls the Format Adjuster to adjust a data output format.

- Start address record output (-entry)

  Outputs the starting address record. The starting address record will be used as a starting address for loading (at the time of debugging).

- Output range (-ran)

  Specifies the range to be adjusted by an address when selecting the option (-adjust) for adjusting an output file.

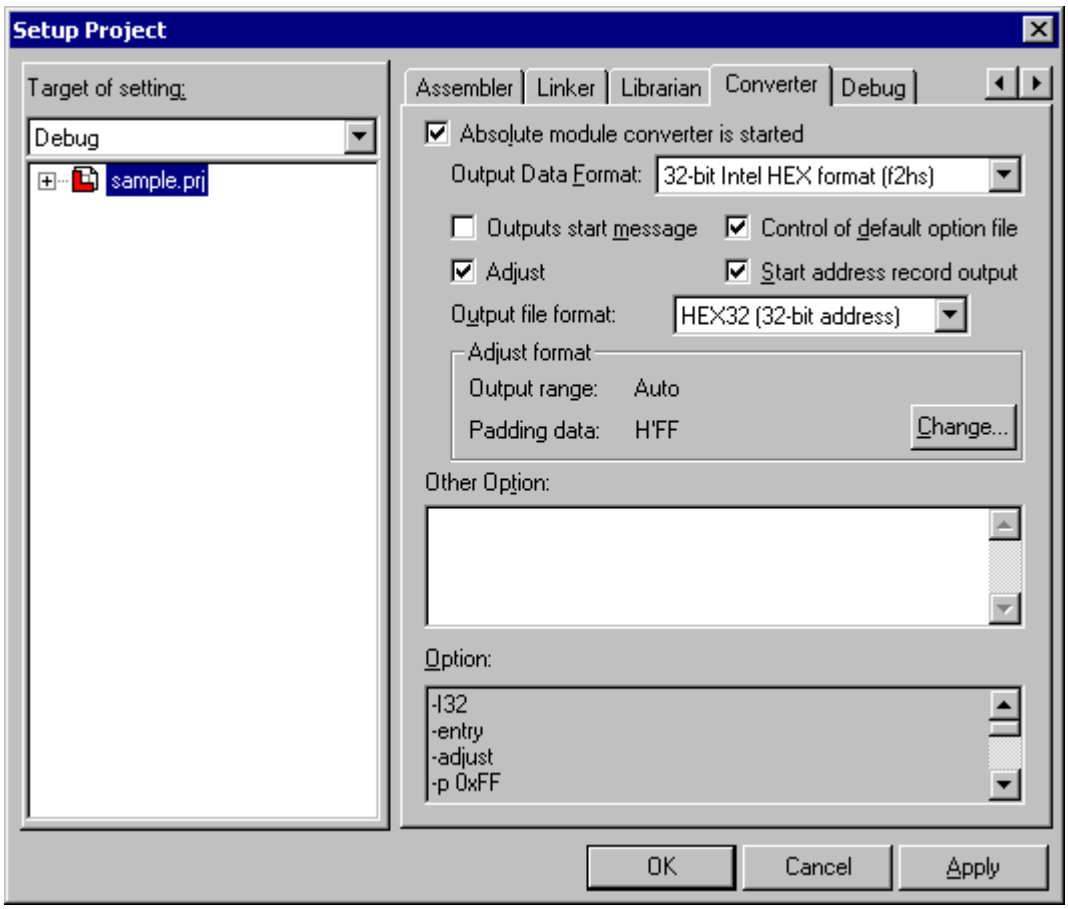
  Selecting [Auto] will obtain the starting/ending address for adjustment from the absolute format load module to set automatically.

- padding data (-p)

  When selecting the option (-adjust) for adjusting an output file, the area of the file where no data exists is packed with data having a specified value.

The macro description can be used to describe options. For the macro description, refer to Section "1.11 Macro Descriptions Usable in Manager" of SOFTUNE WORKBENCH User's Manual.

Figure 4.5-48 Converter Option Setup Dialog Box



4.5.5.9 Setting Debug Options

This section explains how to set debug options.

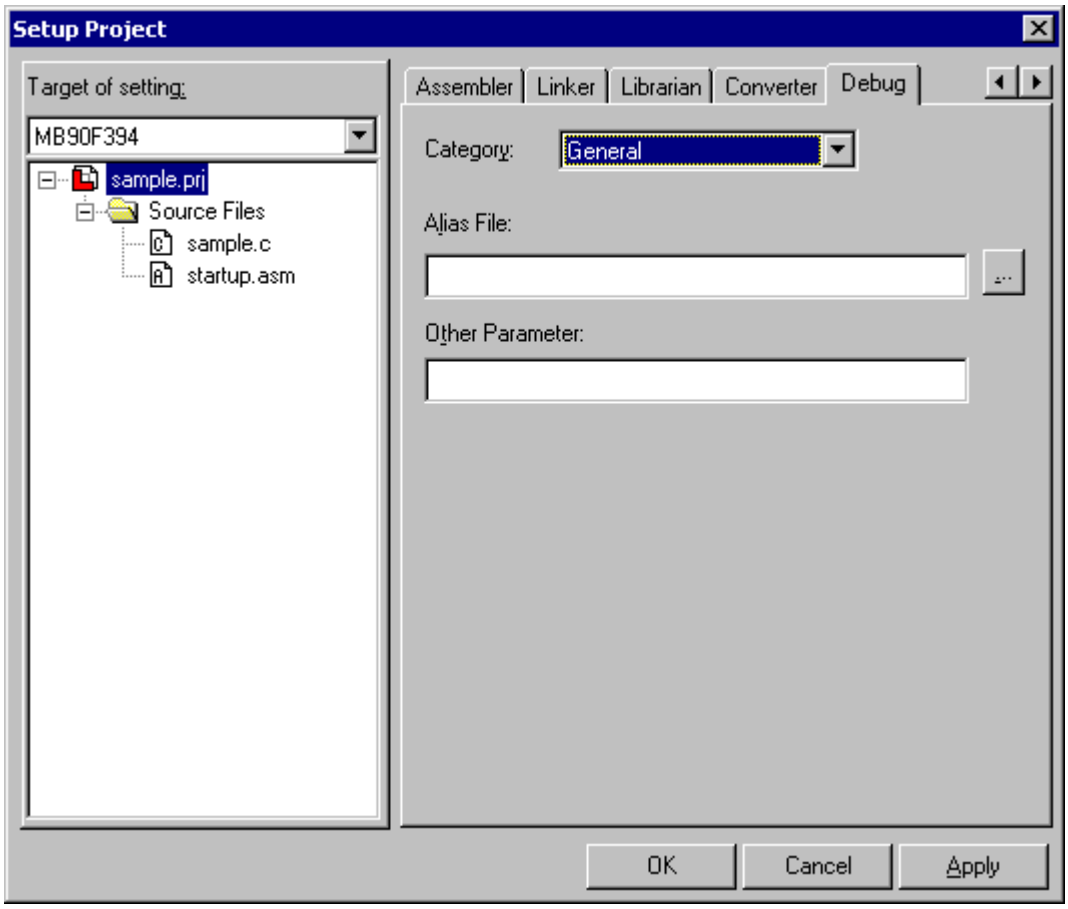
■ Setting debug options

1. Click the [Debug] tab from the project setup dialog box.
   The debug option setup dialog box shown in Figure 4.5-49 opens.
2. Select category.
   [General] or [Setup] can be selected as category.

■ Setting Options in [General]

1. Set an alias file.
   Clicking the [...] button to the right of the input field enables file reference.
2. When setting other options, write them in [Other Parameter].

Figure 4.5-49 Debug Option Setup Dialog Box



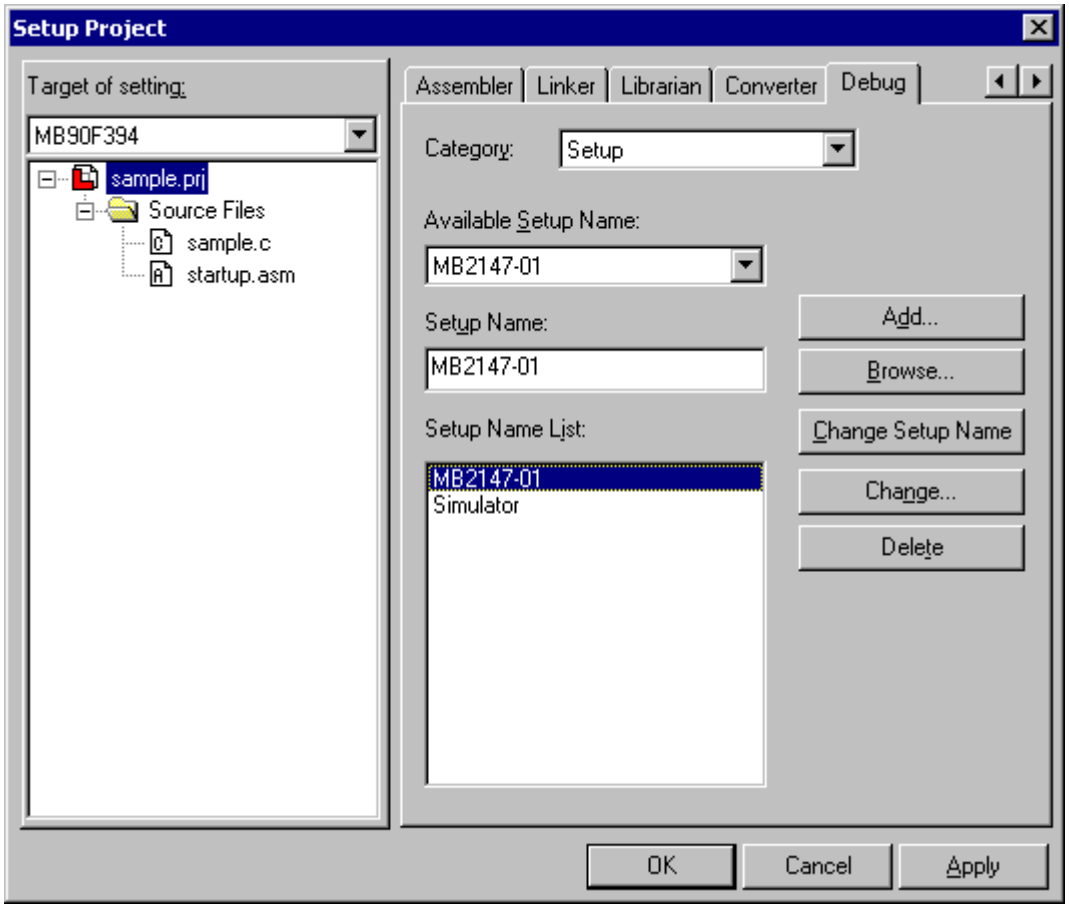
■ Setting Options in [Setup]

1. Click the [Debug] tab from the project setup dialog box.
2. Select [Setup] category.
3. Set [Setup Name].
   Set a different name from registered setup names.
4. Click the [Add Setup] or [Add Reference] button.

   Clicking the [Add Setup] button starts the Setup Wizard and adds the new setup. See Section "4.7.2.5 Setup Wizard".

   Clicking the [Add Reference] button reads information from the set file for setup. When the file selection dialog opens, select a file from the dialog, then click the [Open] button.

Figure 4.5-50 Debug Option Setup Dialog Box



■ Deleting debugger setup

1. Click [Debug] tab from the project setup dialog box.
2. Select [Setup] category.
3. Select the setup name to be deleted from [Setup Name List].
4. Click the [Delete] button.

■ Changing debugger setup

1. Click [Debug] tab from the project setup dialog box.
2. Select [Setup] category.
3. Select the setup name to be changed from [Setup Name List].
4. Click the [Change Setup] button.

   Setup Wizard is started. See Section "4.7.2.5 Setup Wizard".

■ Changing setup name

1. Click [Debug] tab from the project setup dialog box.
2. Select [Setup] category.
3. Select the setup name to be changed from [Setup Name List].
4. Click the [Change Setup Name] button.

4.5.6 Setting Customize Build

This section explains how to start a different tool before or after executing the language tool during Make or Build.

■ Customize Build function

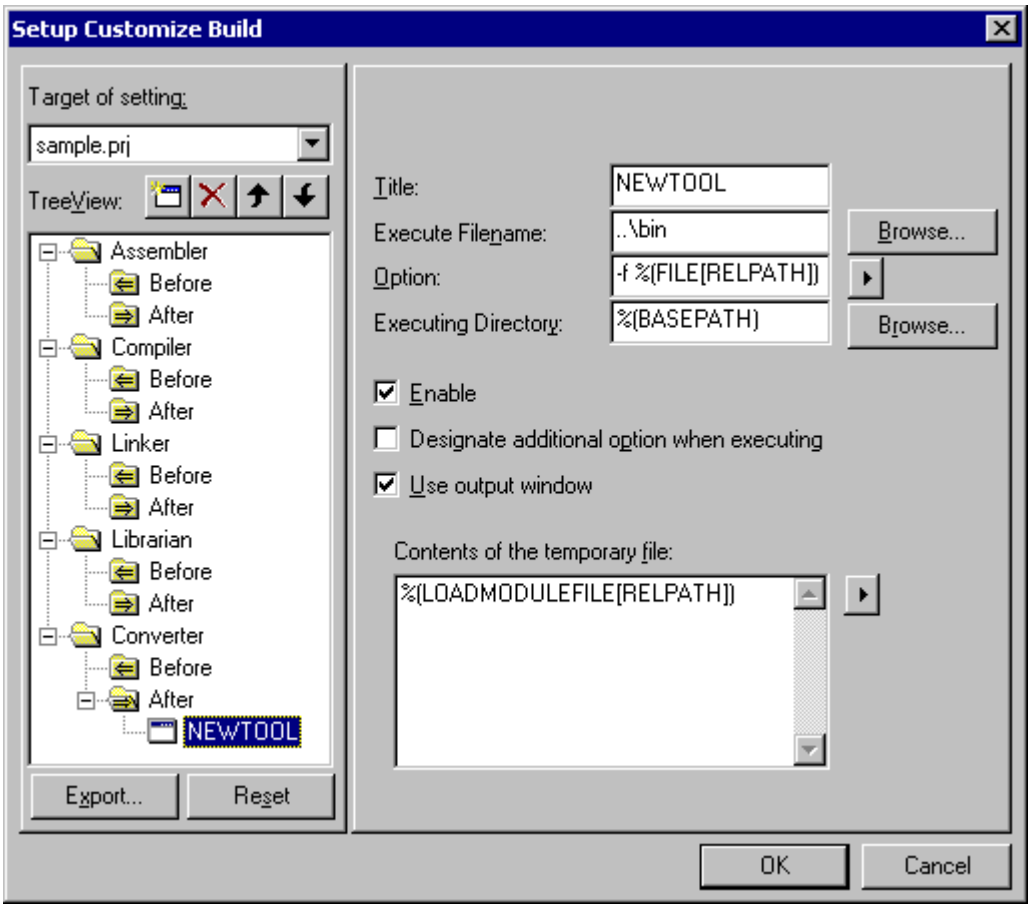
In SOFTUNE WORKBENCH, it is possible to make a different tool operate automatically before or after executing the language tool during Compile, Assemble, Make, or Build. Using this function makes it possible to:

make a customer tool operate before the Compiler is executed.

make the object module conversion tool operate after the Linker is executed.

This setting is stored on a project-by-project basis.

Figure 4.5-51 Setup Customize Build Dialog Box



■ Tool button list

■ Setting Target

There are two types of setting targets in [Target of setting]: Default, and Project name.

See [Target of setting] for which one is currently set.

- Default

  When the Customize Build is set when no project is opened, the default setting can be changed.

  The Customize Build setting is referred when creating a new project and is copied to that project.

  "Default" is displayed in [Target of setting].

Note:

When a project created in an older version is opened, this setting is also referred and copied to that project.

- Project Name

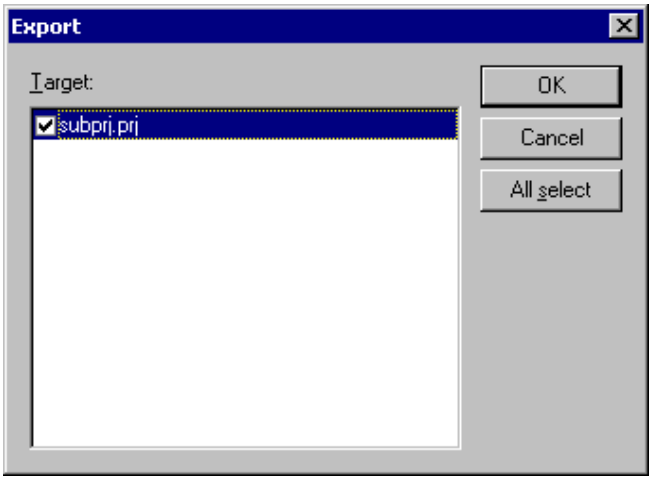
  Customize build is set for the project. The tool operates at Compile, Assemble, Make, or Build for the opened project.

■ Export

The [Export] button is clicked to open the Export dialog shown in Figure 4.5-52.

By specifying the project in this dialog, tool information can be copied to other projects in workspace.

Figure 4.5-52 Export Dialog Box



■ Reset

To clear the currently set state and return to the default setting, click [Reset]. The [Reset] button can be used when the setting target is Project.

■ Title

Input the tool name; duplicated tool names do not cause a problem.

■ Execution File Name

Input the file name of the executed tool.

■ Option

Specify the option for the executed tool. A macro can be specified in this field.

For the macros, refer to Section "1.1.1 in the SOFTUNE WORKBENCH User's Manual".

When the button at the right of this field is clicked, the list of usable macros is displayed. For example, when [Build File]-[Directory] is clicked, %(FILE[PATH]) is inserted for the option at the cursor position.

■ Executing Directory

To execute the tool for a particular directory, specify the Executing Directory. If the Executing Directory is not specified, the tool is executed for the project directory.

■ Enable

This specifies whether or not to execute the tool at Compile, Assemble, Make, or Build. The tool is not executed when the checkbox is unchecked. To suspend execution of the tool, uncheck the checkbox.

■ Designate additional option when executing

This specifies whether or not to display a dialog in which additional options can be specified at executing the tool. When adding options at executing the tool, check the checkbox.

The option set in the specifying other parameter dialog is added as is to the end of the character string specified for [Option].

■ Use Output window

To display the tool execution result in the Output window, check the checkbox. There are some precautions to follow when using the Output window. For details, refer to Section "1.1.1 in SOFTUNE WORKBENCH User's Manual".

■ Contents of the temporary file

When the %(TEMPFILE) macro is specified for [Option], SOFTUNE WORKBENCH creates a new temporary file at executing the tool and deletes it at ending the tool execution. In this field, specify the data to be written to this temporary file. For example, when the tool option becomes very long, it is possible to specify %(TEMPFILE) for [Option] and specify that option in this field (However, this is only valid when the tool to be executed permits specifying the option in the file.)

Macros can be input in this field. For the macros, refer to Section "1.1.1 in SOFTUNE WORKBENCH User's Manual".

■ Setting procedure

1. Select [Project]-[Customize Build].
   - The Customize Build setting dialog is displayed (Figure 4.5-51).
2. Select the tool registered position from the tree view and then press the [NEW] button.
   - When a category is selected and the [NEW] button is pressed, a tool entitled "NEWTOOL" is created at the end of the category. When a tool is selected and the [NEW] button is pressed, a tool entitled "NEWTOOL" is created immediately after the selected tool.
   - Tools in the Before/After category are executed sequentially from the top.
3. Set the title of the tool to be registered.

4. Set the execution file to be registered.
   - When the [Browse] button at the right of this field is clicked, the file selection dialog is displayed and the tool execution file name can be selected from this dialog.
5. Set the option if necessary.
6. Set the execution-time directory if necessary.
   - When the [Browse] button at the right of this field is clicked, the directory selection dialog is displayed and the execution-time directory can be selected from this dialog.
7. Set [Enable], [Designate additional option when executing], and [Use Output window] if necessary.
   - Select a category or two or more tools from the tree view to batch-change [Enable], [Designate additional option when executing], and [Use Output window].
8. Set the temporary file data as necessary.
9. Click the [OK] button to complete setting.

■ Deletion procedure

1. Select [Project]-[Customize Build].
   - The Customize Build setting dialog is displayed (Figure 4.5-51).
2. Use the tree view to select the title of the tool to delete.
   - When a category is selected, all the tools in the category are deleted.
   - Two or more tools can be also selected.
3. Click the [DELETE] button.

■ Start sequence change procedure

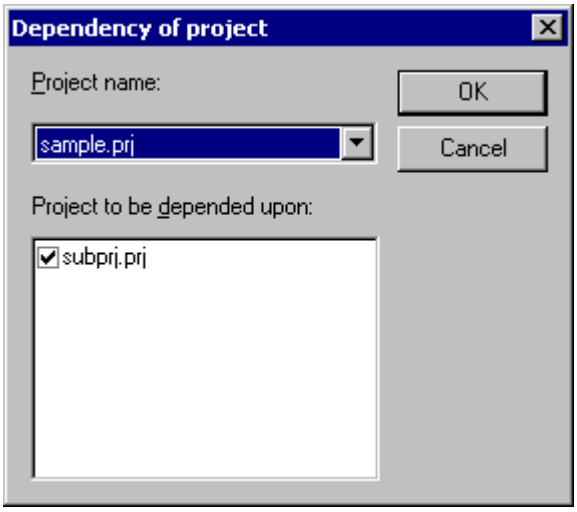
1. Select [Project]-[Customize Build].
   - The Customize Build setting dialog is displayed (Figure 4.5-51).
2. Use the tree view to select the tools for which the start sequence is to be changed.
   - Tools in the Before/After category are sequentially executed from the top.
3. Click the [UP] and [DOWN] buttons to arrange in the start sequence.
   - Tools can be moved only within the category which the tools belong.

4.5.7 Project Dependencies

A subproject is defined in the project.

■ Project Dependencies

Figure 4.5-53 Dependency of Project Dialog Box



- Project name

  The name of the project, which a subproject is defined in or deleted from, is displayed.

- Project to be depended

  The name of the project on which the selected project can depend in "Project name" is displayed.

  The project name indicated by checkmark is the name of the subproject in the project selected in "Project name".

■ Procedure for defining Project dependence

For the procedure to define project dependence, see Section "2.6 Definition of Subproject".

4.5.8 Project Configuration

The project configuration is set.

■ Project Configuration

There are the following menus to set the project configuration:

- Add and Delete

  The project configuration is added and deleted, and the active configuration is changed.

- Configuration at build

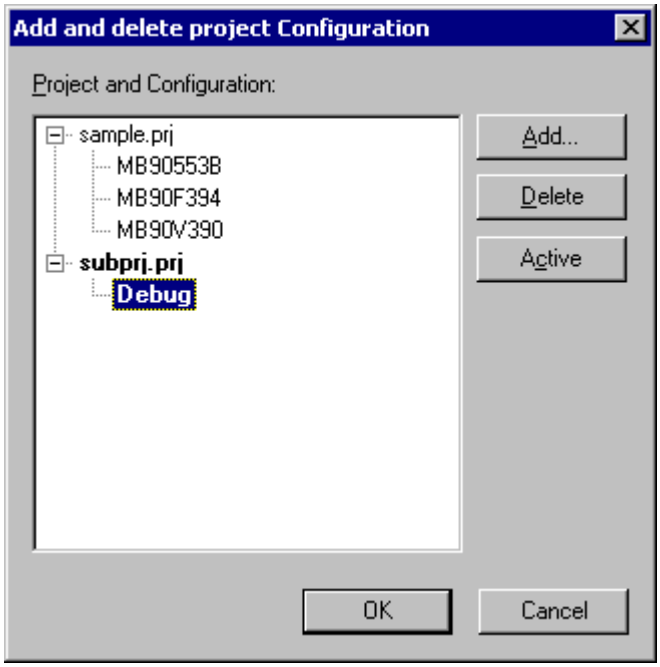
  The configuration as a subproject is made or built is set.

4.5.8.1 Project Configuration - Add and Delete

The configuration as a subproject is made or built is set.

■ Project Configuration Add and Delete

Figure 4.5-54 Add and Delete Project Configuration Dialog Box



- Project and Configuration

  All projects in workspace and their configurations are displayed.

- Add

  Click this button to open the [Add Project Configuration] dialog shown in Figure 4.5-54.

  This dialog enables the addition of the project configuration.

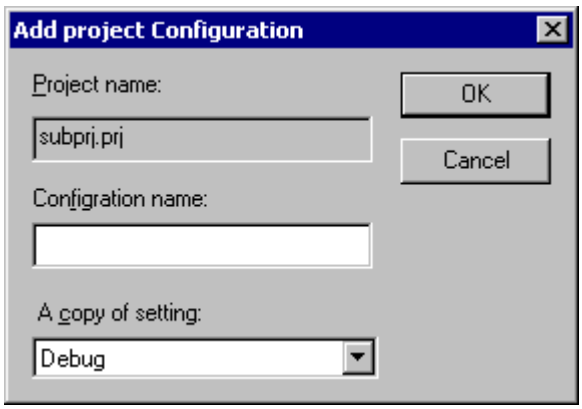
- Delete

  Click to delete the selected project and its configuration.

- Active

  Click this button to make the selected project and its configuration active.

Figure 4.5-55 Add Project Configuration Dialog Box



- Project name

  The name of the project to which the configuration is added is displayed.

- Configuration name

  Set the name of the project configuration to be added.

- A copy of setting

  Select the configuration to which settings are copied.

■ Setting Procedure

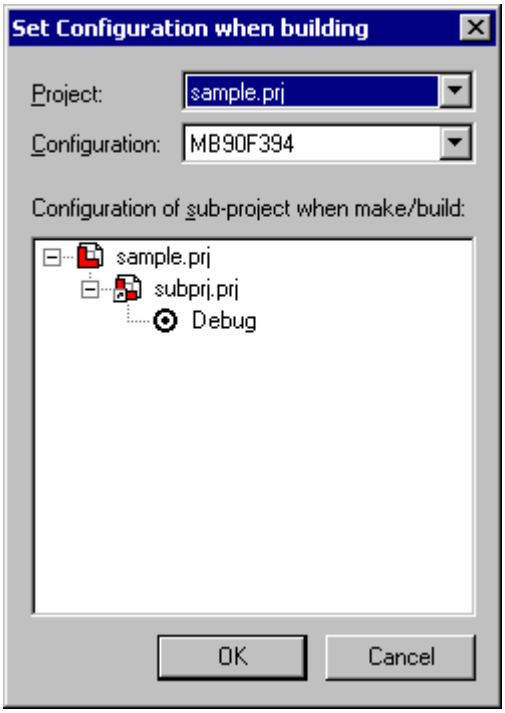
For the setting procedure, see Section "2.7 Creation of Project Configuration".

4.5.8.2 Project Configuration - Configuration at Build

The configuration as a subproject is made or built is set.

■ Project Configuration - Configuration at Build

Figure 4.5-56 Set Configuration when Building Dialog Box



- Project

  Select the parent project of a subproject.

- Configuration

  Select the configuration of the parent project select in "Project."

- Configuration of sub-project when made/built

  Select the configuration of the subproject, as the configuration of the parent project selected in "Configuration" is made or built.

■ Setting Procedure

For the setting procedure, see Section "2.10.1 Making or Building of Project".

4.5.9 Include Dependencies

"Include Dependencies" updates include file dependency.

■ Updating include file dependency

This command (function) checks all the source files in the project file and registers all the include files being used by the source files in the project. The registered include files are displayed in the [Dependencies] category field of the SRC tab of Project Window.

4.5.10 Compile, Make, Build, and Stop

This section explains the functions of Compile, Make, Build, and Stop.

■ Compile

Compile compiles only the specified source file irrespective of whether other source files and include files are corrected. However, compile does not link the specified source file.

This command also assembles the specified file when the file is an assembler source file.

■ Make

Make checks all the source and include files in the project and compiles or assembles only files which have been modified since the last Make or Build.

If some library and object files are modified, Make links them to create a target file.

■ Build

Build compiles or assembles all the source files in the project regardless if they have been modified since the last Make or Build.

It also links all object and library files to create a target file.

■ Stop

Use Stop when you want to stop compile, make, or build execution for some reason.

■ Checking source file saving

When an unsaved file is being edited by the standard editor, execute compile (assemble), make, or build, then save the file. If a check mark is set to the left of [Inquiry for Save at Compile/Assemble] in [Setup]-[Development...]-[Project], however, the dialog box for asking whether to save the file opens. When the [No] button is clicked, the source file is compiled without being saved.

4.6 Debug

"Debug" starts and terminates debugging and controls the debugger when SOFTUNE WORKBENCH is in the debug session.

■ Debug start and termination

-Loading Target File

-Start Debug/Terminate Debug

■ Debugger control when SOFTUNE WORKBENCH is in the debug session

-Run

-Abort

-Reset MCU

-Breakpoints...

-Breakpoint Set/Reset

-Event...

-Sequence

-Stack...

-Time Measurement...

-Call...

-Clear Call

-Vector

4.6.1 Run

This section explains the debugger program execution function.

■ Run

"Run" provides the following six functions:

-Go

When [Go] is clicked, the debugger continuously executes the program from the current PC position. When a breakpoint is reached or when [Abort] is selected from the [Debug] Menu, the debugger stops program execution.

-Step In

When [Step In] is clicked, the debugger executes the step, moves the PC to the address of the next instruction, and stops. When a function call instruction is executed, the debugger stops at the beginning of the function.

-Step Over

When [Step Over] is clicked, the debugger executes the step, moves the PC to the beginning of the next instruction, and stops. When a function call instruction is executed, the debugger executes all the functions, moves the PC to the next instruction address of the function call instruction, and stops.

-Step Out

When [Step Out] is clicked, the debugger executes the current function to the end, returns control to the function caller, moves the PC to the next instruction address of the function call instruction, and stops.

-Run Until Cursor

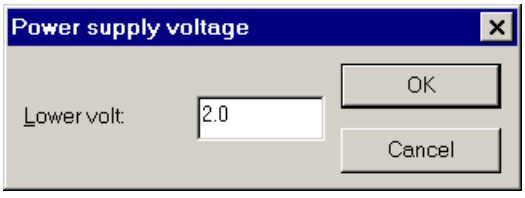
When [Run Until Cursor] is clicked, the debugger executes the program to the instruction immediately before the address indicated by the cursor (in the Source or Assembly Window), moves the PC to the address, and stops.

-Power On Debug

When starting Power-on-Debug, the [Power supply voltage] dialog is opened.

It is able to set and to check the Lower-limit-voltage.

Figure 4.6-1 Power Supply Voltage



4.6.2 Abort

This section explains the debugger program execution stop function.

■ Abort

[Abort] is used to forcibly interrupt the program being executed by the debugger. When the program stops, the PC moves to the next instruction address of the last executed instruction. Source line display and disassemble display are also updated according to the PC value set when the program stopped.

Note:

The monitor debugger does not provide this stop (abort) function.

4.6.3 Reset MCU

This section explains the MCU reset function of the debugger.

■ MCU reset function

The MCU reset function resets the MCU.

-Emulator debugger

The MCU reset function issues the reset signal to the emulator.

-Simulator debugger

As with the actual chip, set the initial values of the registers to be initialized by reset and clear other registers to 0.

In reset MCU, breakpoints, watch points, map setting, and program variables are not modified.

Note:

The monitor debugger does not provide the MCU reset function.

4.6.4 Break Point

This section explains how to set, reset, and modify a breakpoint and how to display a break list.

■ Breakpoint

When the PC passes an address or the program accesses data at an address, the position where program execution by the debugger is to be stopped is called a breakpoint.

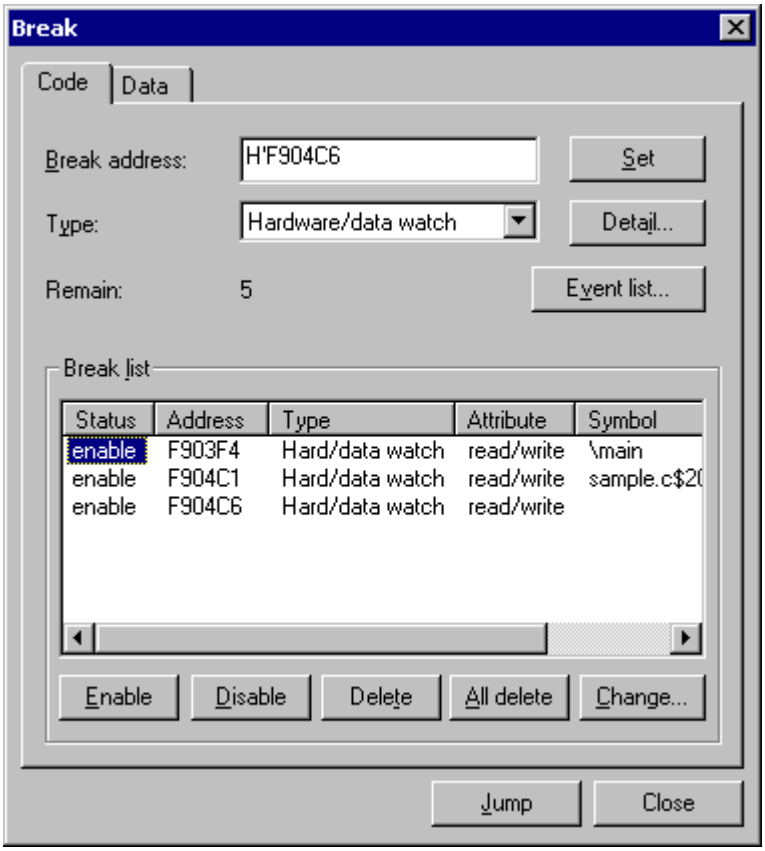
■ Code breakpoint

When the PC passes the set address (when the address is executed), the breakpoint where program execution is to be stopped is called a code breakpoint.

■ Setting of code breakpoint

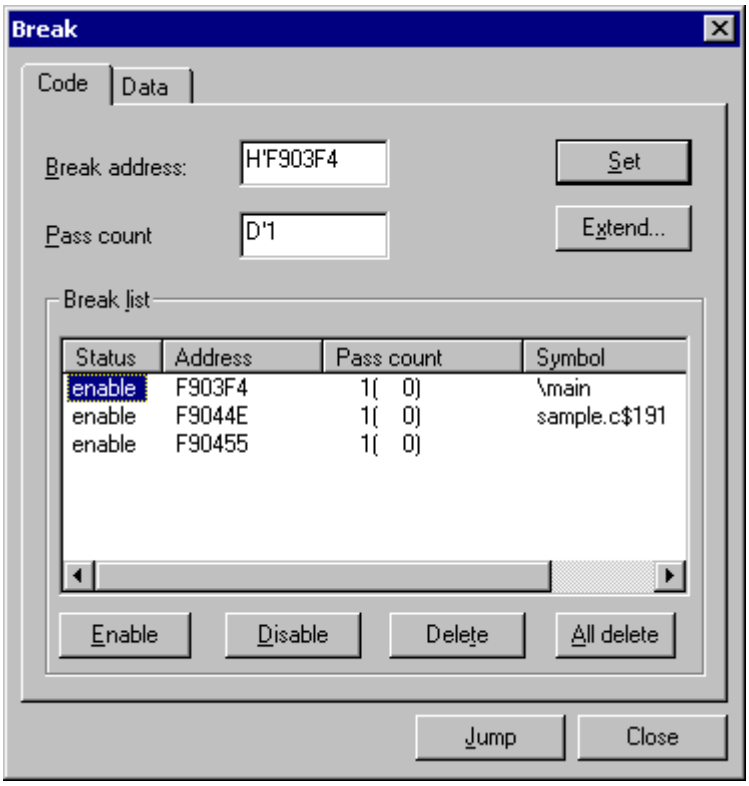
[Emulator Debugger]

Figure 4.6-2 Break Dialog Box (Code)



[Simulator Debugger]

Figure 4.6-3 Break Dialog Box (Code)



- Break address

  In this field, the address that sets a breakpoint is specified.

- Type

  In this field, the type of the breakpoint is selected. This field is only enabled for MB2147-01.

- Remaining count

  In this field, the remaining count of settable breakpoint types currently being selected is displayed. Regarding the hardware breakpoints for MB2147-05 and MB2147-01, this field is displayed when the remaining count of settable breakpoint types has become 10 or fewer. Also, this field is not displayed for some debugger types.

- Pass count

  In this field, the count of times the PC passes that point (a particular point) before causing a break is set. This field is only enabled for the simulator.

- Breakpoint list

  A list of code breakpoints currently being set is displayed.

State	:	Enable or disable is displayed.
Type	:	The breakpoint type is displayed.
Address	:	The set address is displayed.
Attribute	:	The breakpoint attribute is displayed.
Pass count	:	The set pass count is displayed. In ( ), the count of times the PC has passed that point by the present time is displayed.

Symbol : The symbol or the line number of the source file assigned to the address are displayed.

- [Setting] button, [Add] button

  These buttons are used to set a breakpoint at the specified address. When an address that is already set in the breakpoint list is specified, the set data of the breakpoint at this address is changed.

- [Details] button

  This button is used to set the details of a breakpoint at the specified address. This button is only enabled for MB2147-01. For details, see Section "4.6.4.1 Breakpoint Details Setting".

- [Event List] button

  This button is used to display the event list for checking the setting of all events. This button is enabled for MB2147-01 when "Monitor hardware/data" is selected for breakpoint type.

- [Debug Area] button

  This button is used to set the debug area needed at the time of breakpoint setting. This button is enabled for MB2147-05.

- [Enable] button

  This button is used to enable the breakpoints in the breakpoint list currently being selected.

- [Disable] button

  This button is used to disable the breakpoints in the breakpoint list currently being selected. The breakpoints are simply disabled; that is, the setting itself of the breakpoints is not cancelled.

- [Delete] button

  This button is used to delete the setting of the breakpoints in the breakpoint list currently being selected.

- [Delete All] button

  This button is used to delete the setting of all the breakpoints in the breakpoint list.

- [Change] button

  This button is used to change the setting of the breakpoints in the breakpoint list currently being selected.

- [Jump] button

  Moves a starting position for displaying the source window to the code position at the breakpoint selected from the break list.

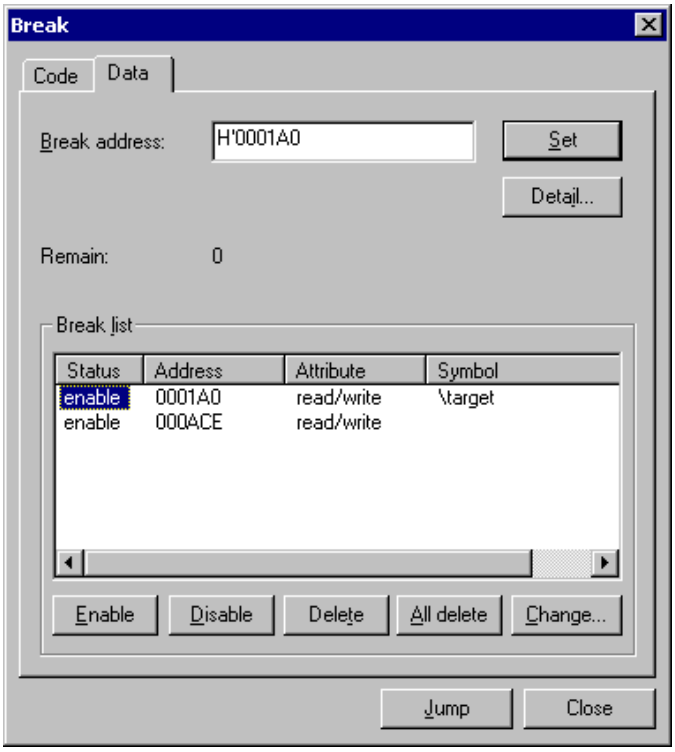
■ Data breakpoint

Such a breakpoint as stop the program when data at the set address is accessed is called the data breakpoint.

■ Setting of data breakpoint

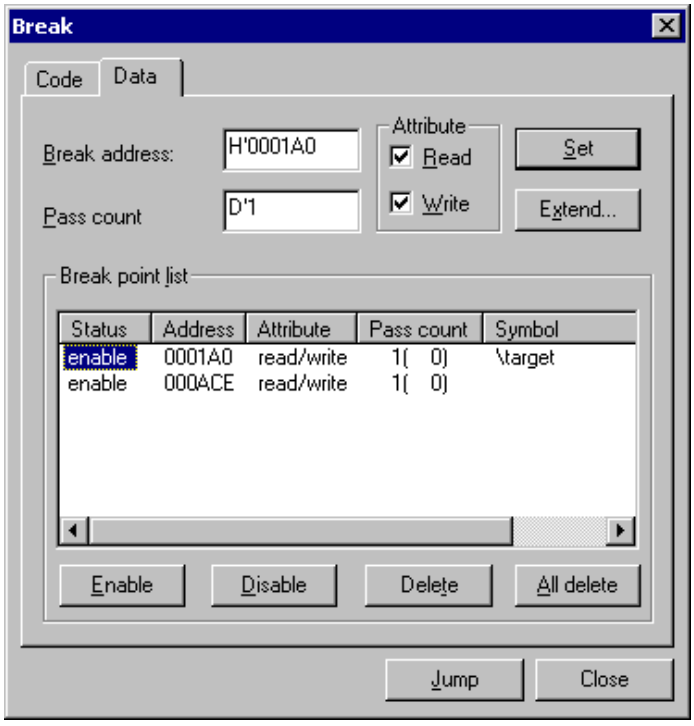
[Emulator Debugger]

Figure 4.6-4 Break Dialog Box (Data)



[Simulator Debugger]

Figure 4.6-5 Break Dialog Box (Data)



- Breakpoint address

  In this field, the address that sets a breakpoint is specified.

- Remaining count

  In this field, the remaining count of settable breakpoint types currently being selected is displayed.

- Pass count

  In this field, the count of times access with the specified attribute is to be made to that point (a particular point) before causing a break is set. This field is only enabled for the simulator debugger.

- Attribute

  In this field, read access, write access, or both is specified.

- Breakpoint list

  A list of data breakpoints currently being set is displayed.

State	:	Enable or disable is displayed.
Address	:	The set address is displayed.
Attribute	:	The attribute is displayed.
Pass count	:	The set pass count is displayed. In ( ), the count of times the PC has passed that point by the present time is displayed.
Symbol	:	The symbol given to that address is displayed.

- [Setting] button, [Add] button

  These buttons are used to set a breakpoint at the specified address. When an address that has been already set in the breakpoint list is specified, the set data of the breakpoint at this address is changed.

- [Details] button

  This button is used to set the details of a breakpoint at the specified address. For details, see Section "4.6.4.1 Breakpoint Details Setting".

- [Enable] button

  This button is used to enable the breakpoints in the breakpoint list currently being selected.

- [Disable] button

  This button is used to disable the breakpoints in the breakpoint list currently being selected. The breakpoints are simply disabled; that is, the setting itself of the breakpoints is not cancelled.

- [Delete] button

  This button is used to delete the setting of the breakpoints in the breakpoint list currently being selected.

- [Delete All] button

  This button is used to delete the setting of all the breakpoints in the breakpoint list.

- [Change] button

  This button is used to change the setting of the breakpoints in the breakpoint list currently being selected.

- [Jump] button

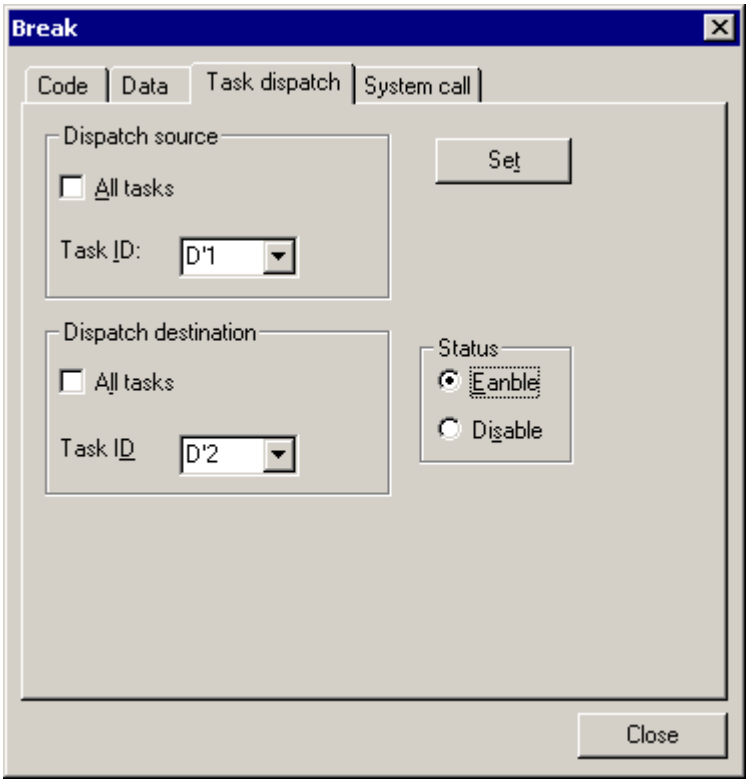
  Displays data at the breakpoint selected from the break list in the memory window.

■ Task dispatch break

Note:

The task dispatch break is displayed only when REALOS is valid.
The breakpoint where the program is stopped when the specified task is dispatched to another task is called the task dispatch break.

Figure 4.6-6 Break Dialog Box (Task Dispatched Break)



- Dispatch Source
  - All Tasks

    Set a check mark to the left of this field when specifying all tasks.
  - Task ID

    Specify the task ID of the task dispatch source.
- Dispatch Destination
  - All Tasks

    Set a check mark to the left of this field when specifying all tasks.
  - Task ID

    Specify the task ID of the task dispatch destination.
  - Status

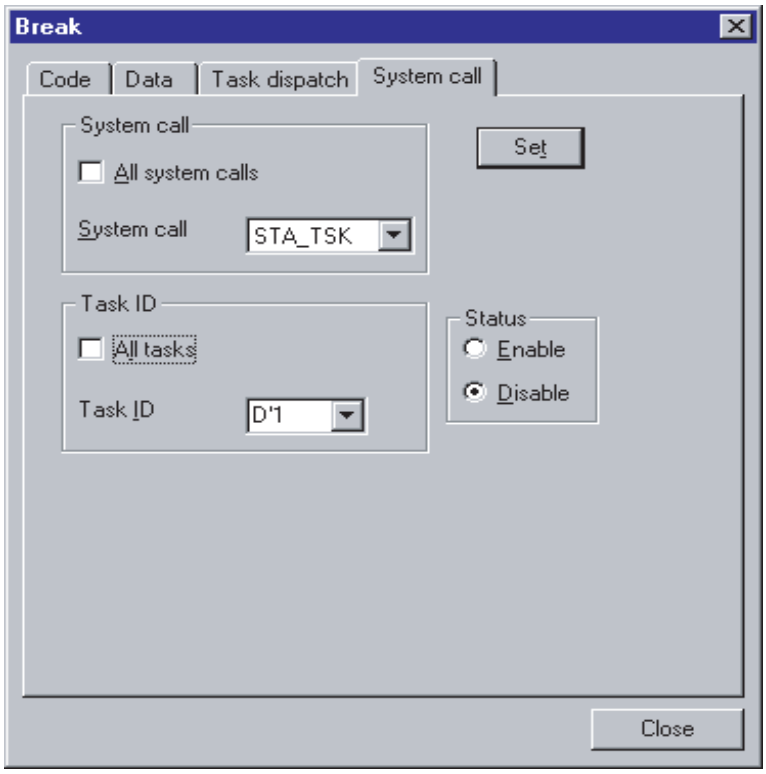
    Specify whether to enable or disable the task dispatch break.

■ System call break

Note:

The system call break is displayed only when REALOS is valid.
The breakpoint where the program is stopped when the specified system call is issued to the task is called the system call break.

Figure 4.6-7 Break Dialog Box (System Call Break)



- System Call
  - System Call Name

Specify the system call name to be monitored.

- Task ID
  - All Tasks

    Set a check mark to the left of this field when specifying all tasks.
  - Task ID

    Specify the task ID to be monitored.
- State

  Specify whether to enable or disable the system call break.

■ Setting and resetting a breakpoint

In the Source or Assembly Window, a breakpoint can be easily set at the address indicated by the cursor. The breakpoint set at the address indicated by the cursor can be also reset easily. Set and reset this breakpoint as follows:

-Select [Breakpoint Set/Reset] from the [Debug] Menu.

The breakpoint is alternately set and reset each time [Breakpoint Set/Reset] is selected.

-Click the left button of the mouse in the breakpoint display field of each window.

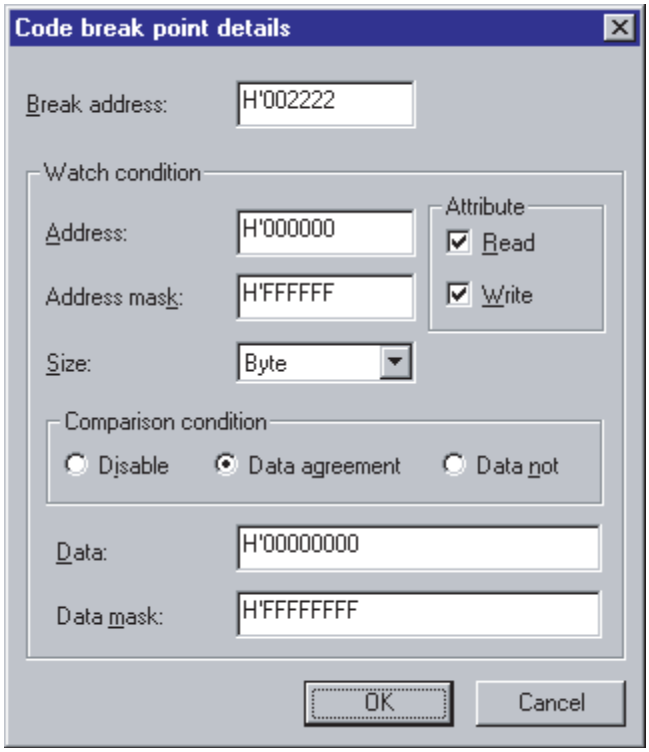
The breakpoint is alternately set and reset each time the left button is clicked.

4.6.4.1 Breakpoint Details Setting

Setting of breakpoint details is explained here.

■ Details of code breakpoint: Debug type (hardware/data monitoring); for MB2147-01

Figure 4.6-8 Code Breakpoint Details Dialog (Hardware / Data Monitoring)



- Breakpoint address

  In this field, the address that sets a breakpoint is specified.

- Address

  In this field, the address or symbol for which data monitoring is to be performed is specified.

- Address mask

  In this field, the mask value for the specified address is specified.

- Size (byte/word)

  The data size at the time of data access is specified.

- Attribute

  The attribute at the time of data access is specified.

- Comparison condition

  The data comparison condition is specified.

Disable: No data is specified for breakpoint condition.

Data match: Data match (data matches the specified data) is specified for the breakpoint condition.

Data NOT: Data mismatch (data does not match the specified data) is specified for the breakpoint condition.

- Data

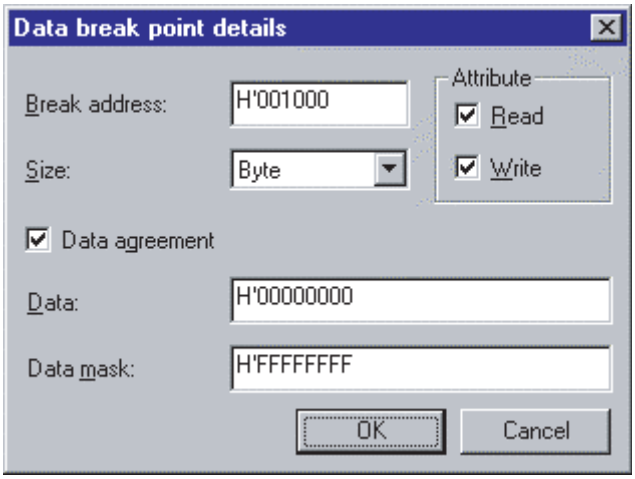
In this field, data at the time of data access is specified.

- Data mask

In this field, the mask value for the specified data is specified.

■ Details of data breakpoint: (For MB2147-01, MB2147-05)

Figure 4.6-9 Data Breakpoint Details Dialog



- Breakpoint address

In this field, the address that sets a breakpoint is specified.

- Size (byte/word)

The data size at the time of data access is specified.

- Attribute

The attribute at the time of data access is specified.

- Data agreement

When specifying the data and the data mask, enter the check mark in the box for "Data agreement".

- Data

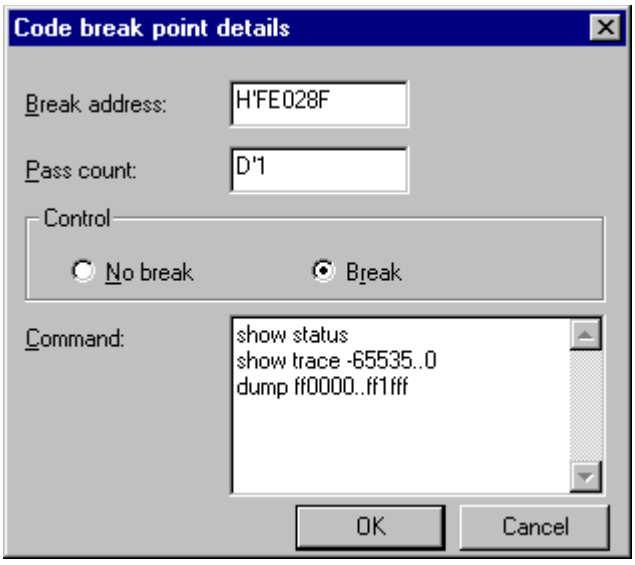
In this field, data at the time of data access is specified.

- Data mask

In this field, the mask value for the specified data is specified.

■ Code Break Point Details (Simulator Debugger)

Figure 4.6-10 Code Break Point Details Dialog (Simulator Debugger)



- Break address

  This field specifies the address to which set a break point.

- Pass count

  This field specifies how many times PC needs to pass there to break.

- Control

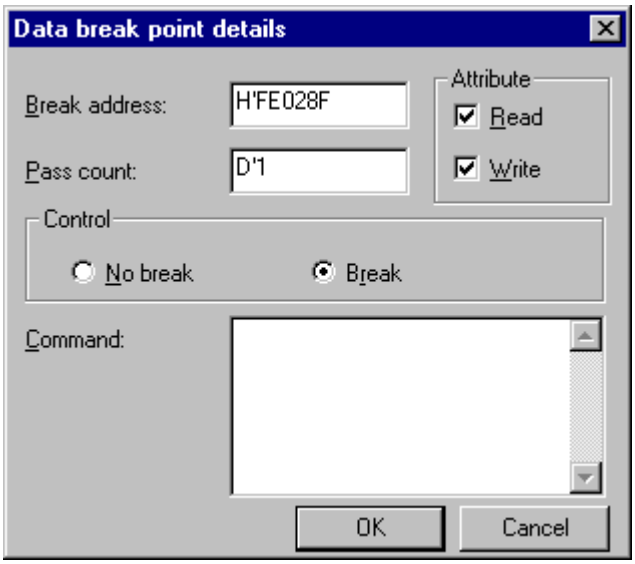
  Specifies whether to re-execute or stop execution after the command processing when the break point is hit.

- Command

  Specifies the command line to execute when the break point is hit.

■ Data Break Point Details (Simulator Debugger)

Figure 4.6-11 Data Break Point Details Dialog (Simulator Debugger)



- Break address

  This field specifies the address to which set a break point.
- Pass count

  This field specifies how many times PC needs to pass there to break.
- Attribute

  Specifies the attributes when data accessing.
- Control

  Specifies whether to re-execute or stop execution after the command processing when the break point is hit.
- Command

  Specifies the command line to execute when the break point is hit.

Note:

For more information about STUB function, refer to Section "2.1.7 STUB Function" in SOFTUNE Workbench User's Manual.

4.6.5 Event

This section explains how to set SOFTUNE WORKBENCH events.

■ Setting events

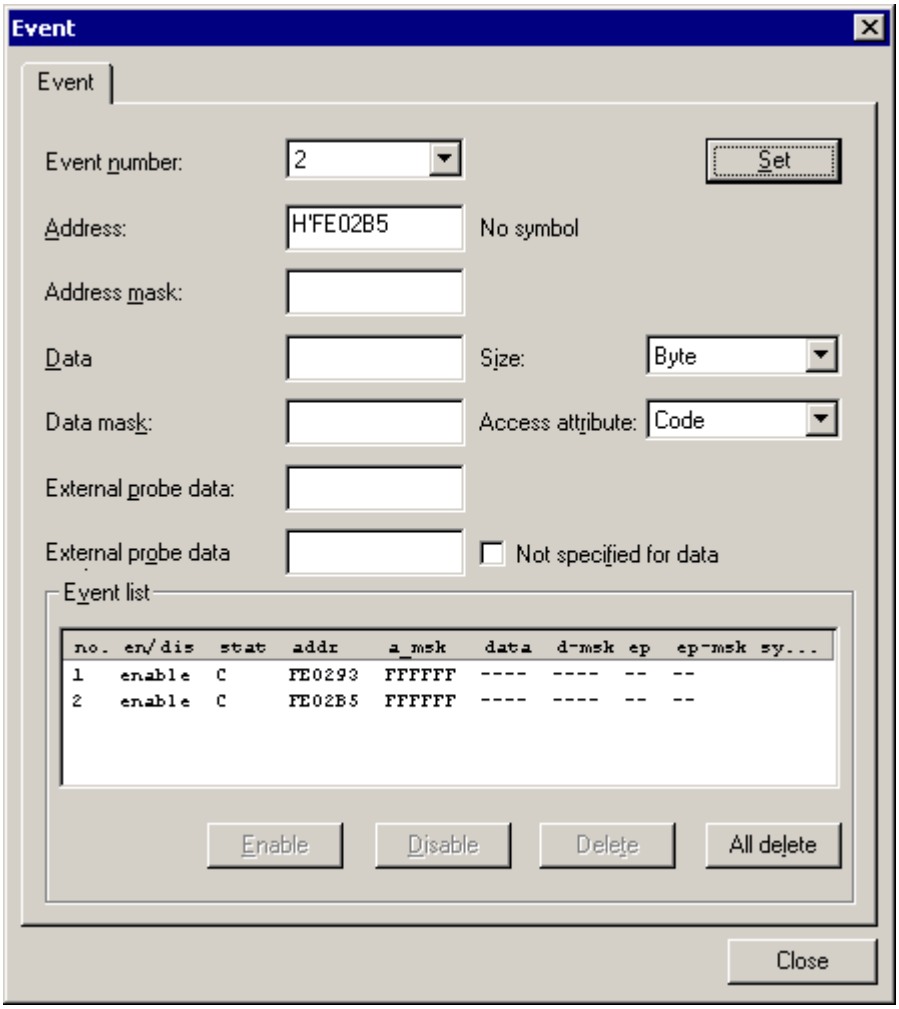
Events can be set from the event dialog box shown in Figure 4.6-12.

. This function can be used only in the emulator debugger.

- Event

[MB2141 (event mode: Normal)]

Figure 4.6-12 Event Dialog Box (Event)



- Event Number

Specifies an event number (1 to 8).

- Address

Specifies the address at which the event occurrence condition is to be set.

- Address Mask

  Specifies address mask. Only the addresses whose bits are 1 are to be compared.

- Data

  Specifies the data to be set as the event occurrence condition.

- Data Mask

  Specifies data mask. Only the data items whose bits are 1 are to be compared.

- Data Length

  Specifies the data length in bytes or words.

- Access Attribute

  Specifies a data access attribute. (Code/Read/Write/Read Code/Read Write/Modify)

- External Probe Data

  Specifies the external probe data value to be set as the event occurrence condition.

- External Probe Data Mask

  Specifies external probe data mask. Only the data items whose bits are 1 are to be compared.

- Not specified for data

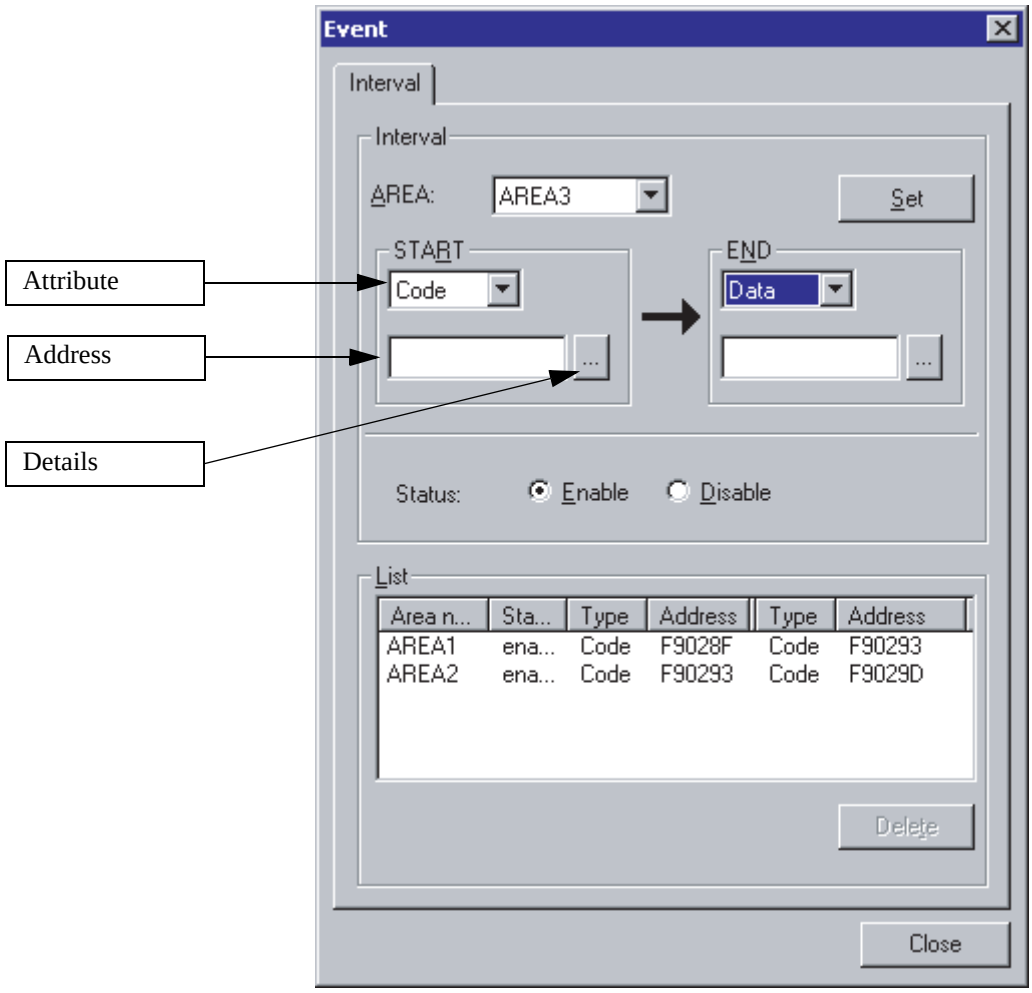
  Specifies the condition when the data values do not match.

- Event List

  Displays the current event setting state.

[MB2147-01 (event mode: performance)]

Figure 4.6-13 Event Dialog Box (Event)



- START/END

  Enter both the start condition and the end condition.

- Attribute

  Specify the attributes of the start/end condition. Select code or data as an attribute.

- Address

  Specify the addresses or symbols to set the start/end condition.

- Details

  Set the details of the start/end condition. Clicking the button, the event details setting dialog box appears.

Figure 4.6-14 Event Details Dialog

- Address

  Specify the addresses or symbols to set the start/end condition.

- Address mask

  Specify the mask value for the specified address.

- Size (byte/word)

  Specify the data size of data-access-time.

- Attribute

  Specify the attribute of data-access-time.

- Conditions for comparison

  Specify condition for data comparison.

  - Invalid

    Data is not specified for the transition condition.

  - Data agreement

    The case where the data matches the specified data is the start/end condition.

  - Data not

    The case where the data does not match the specified data is the start/end condition.

- Data

  Specify the data-access-time data.

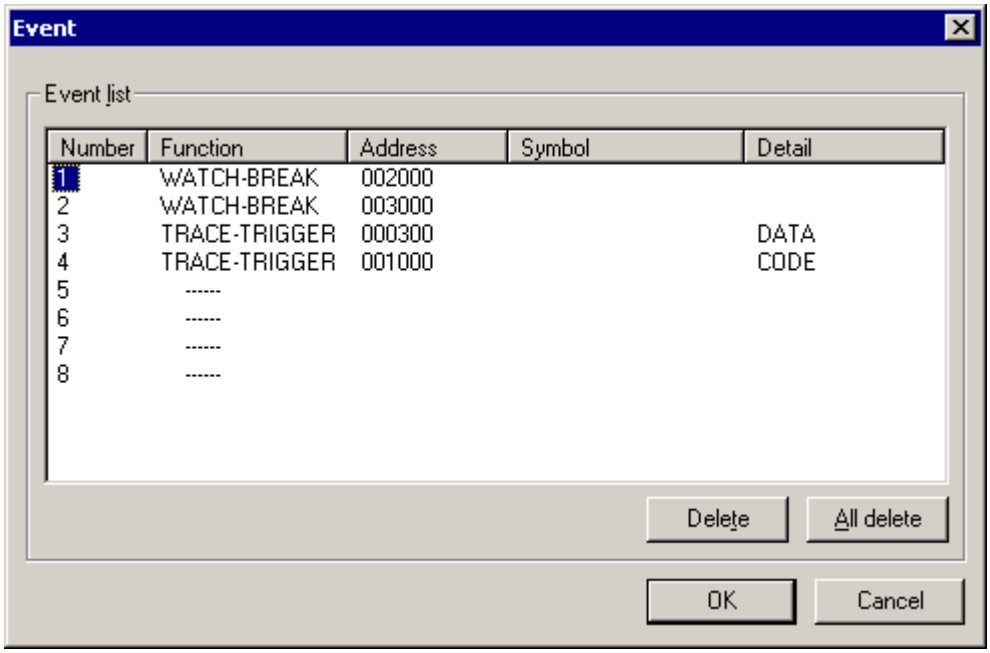
- Data mask

  Specify the mask value for the specified data.

■ Event List

A list of events currently being set is displayed. This function can only be used for MB2147-01.

Figure 4.6-15 Event List Dialog Box



- Event list

  In the event list, events currently being set are displayed.

- [Delete] button

  This button is used to delete the setting of the events in the event list currently being selected.

- [Delete All] button

  This button is used to delete the setting of all the events in the event list.

4.6.6 Sequence

"Sequence" displays the Sequence Window (MB2141) and the Sequence dialog (MB2147-01).

■ Sequence Window

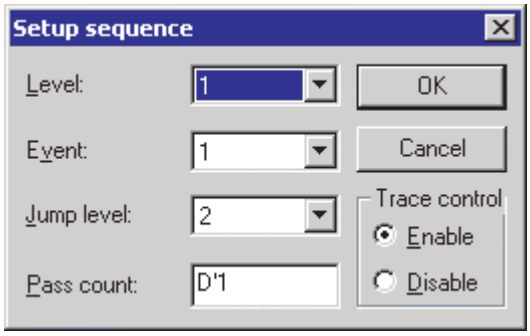
The window or the dialog that displays the sequence setting state opens. A sequence, latch, and delay count can be set from this window.

This function can be used in the emulator.

■ Setting the sequence (MB2141)

Select [Setup] from the shortcut menu of the Sequence Window.

Figure 4.6-16 Sequence Setting Dialog

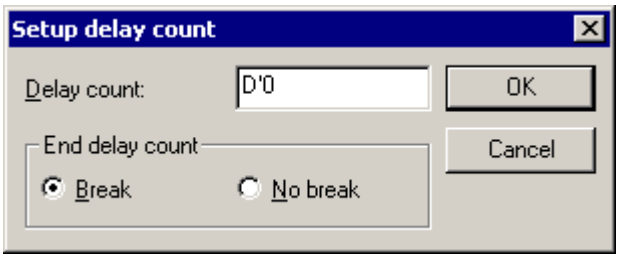


-Level	Select a level.(1-8)
-Event	Select an event.(1-8/Timer)
-Jump Destination Level	Select a jump destination level. (1-8/END)
-Pass Count	Set a pass count.
-Trace Control	Select "Enable" or "Disable".

■ Setting the delay count (MB2141)

Select [Delay Count] from the shortcut menu of the Sequence Window.

Figure 4.6-17 Delay Count Setup Dialog Box

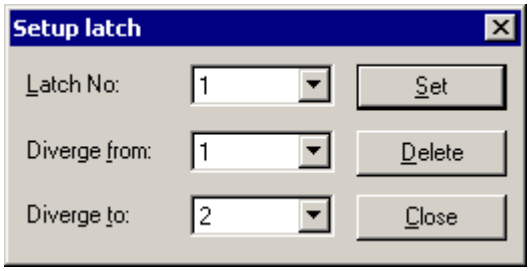


-Delay Count	Set the delay count.
-Delay Count Break	Select "Break" or "Not Break".

■ Setting latch (MB2141)

Select [Setup Latch] from the shortcut menu of the Sequence Window.

Figure 4.6-18 Latch Setup Dialog Box



-Latch Number Select a latch number. (1-2)

-Diverge from Select a branching source level. (1-8)

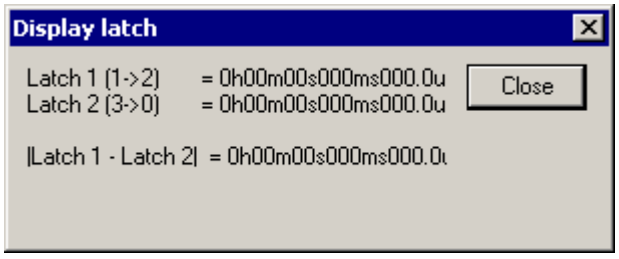
-Diverge to Select a branching destination level. (1-8/END)

-Delete Deletes the set items.

■ Displaying the latch measurement result (MB2141)

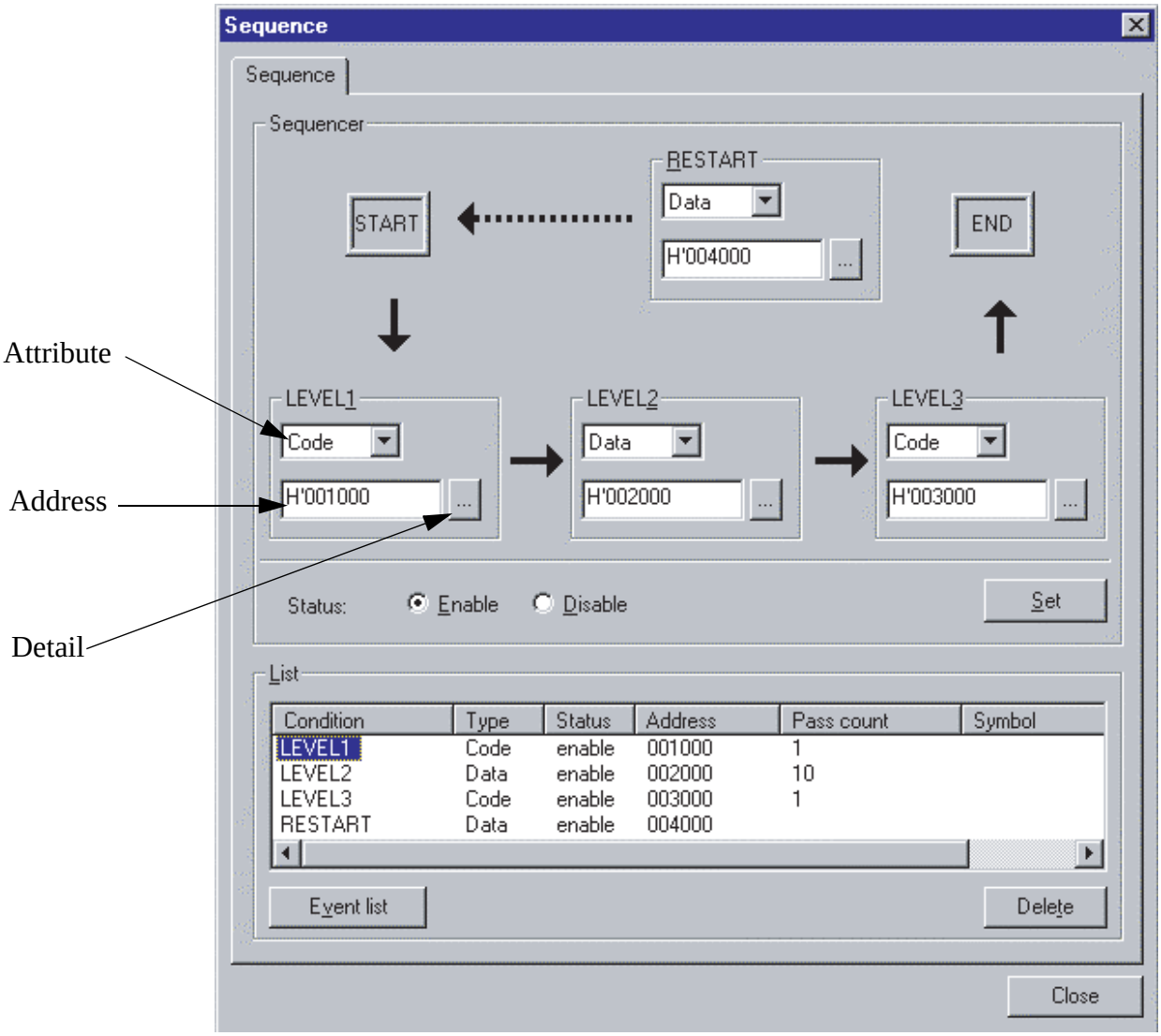
Select [Latch Display] from the shortcut menu of the Sequence Window. The latch measurement result is displayed.

Figure 4.6-19 Latch Display Dialog Box



■ Setting the sequence (MB2147-01)

Figure 4.6-20 Sequence Setting Dialog



-Sequencer

The transition condition for the sequencer is set. A maximum of 3 levels from LEVEL1 to LEVEL3 can be set. Also, RESTART to return control to the starting state of the sequencer can be set. Setting must be performed in sequence starting with LEVEL1. When correct setting is not performed or when the setting is deleted, the subsequent setting will be entirely deleted.

Attribute

The attribute of the transition condition is specified. The attribute is selected from between code and data.

Address

The address or symbol for which the transition condition is to be set is specified.

Details

The details of the transition condition are set. The dialog given in Figure 4.6-21 is displayed.

-State control

Enabling or disabling of the set sequencer is set.

- [Set] button

The transition condition for the specified sequencer is set.

-Setting

The transition condition currently being set for the sequencer is displayed.

- [Event List] button

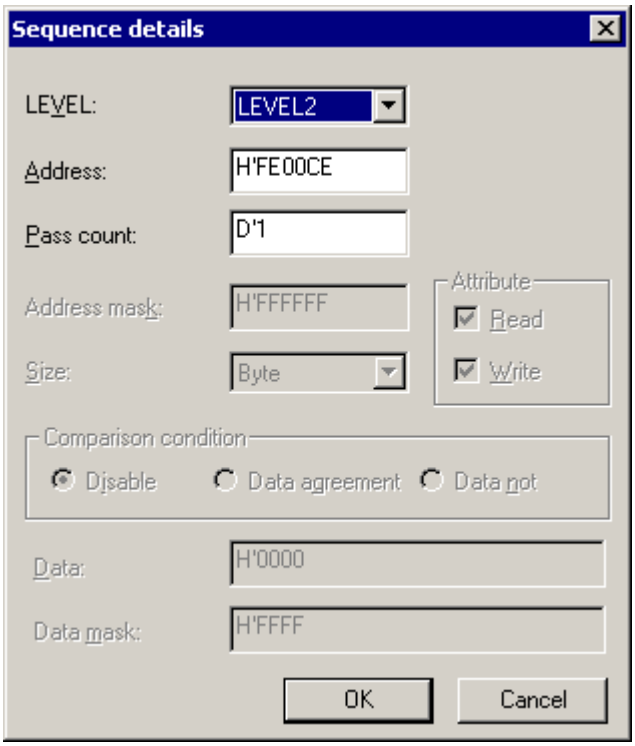
This button is used to display the event list dialog used to check the setting of all events.

- [Delete] button

This button is used to delete the setting.

■ Sequence details setting (MB2147-01)

Figure 4.6-21 Sequence Setting Dialog



-LEVEL

LEVEL for which the transition condition is to be set is set.

-Address: The address or symbol for which the transition condition is to be set is specified.

-Pass count

The access count at the time of trigger hit is specified.

-Address mask

The mask value for the specified address is specified.

-Size (byte/word)

The data size at the time of data access is specified.

-Attribute

The attribute at the time of data access is specified.

-Comparison condition

The data comparison condition is specified.

Disable

No data is specified for the transition condition.

Data match

Data match (data matches the specified data) is specified for the transition condition.

Data NOT

Data mismatch (data does not match the specified data) is specified for the transition condition.

-Data

In this field, data at the time of data access is specified.

-Data mask

In this field, the mask value for the specified data is specified.

4.6.7 Stack

This section explains a SOFTUNE WORKBENCH call stack.

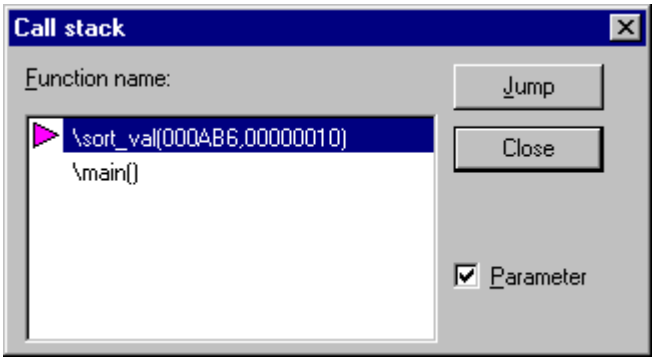
■ Call stack

Usually, a program is a set of several subroutines. For this reason, as debugging advances, function calls of several stages occur. For example, one routine calls another and the called routine calls further another.

The call stack retains the relationship among function calls. Clicking a function name from the function name list immediately displays information for the function in the Source Window.

Next to a function name is displayed a triangular symbol.

Figure 4.6-22 Call Stack Dialog Box



The function written in the lowermost line of the function name list is the main function. This main function calls the function above it. The called function calls further a function above it. In this way, the function written in the uppermost line is the function in which the current PC exists.

When return is executed, functions are deleted in turn from the function name list, starting from the uppermost line.

-Argument Display

When a check mark is set to the left of Argument Display, an argument value is displayed after each function name, as shown in Figure 4.6-22.

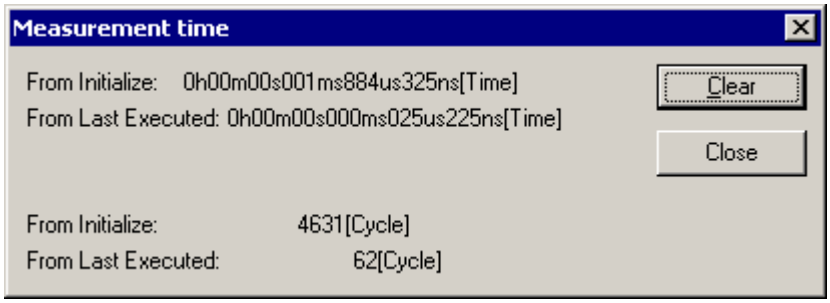
When no check mark is set to the left of Argument Display, only parentheses "( )" are displayed after each function name.

4.6.8 Time Measurement

This section explains SOFTUNE WORKBENCH time measurement.

■ Time measurement

Figure 4.6-23 Time Measurement Dialog Box



-Items to be displayed

	Time	Cycle count	Step count
SIM	X	❍	❍
EML(MB2141)	❍	X	X
EML(MB2147-01)	❍	❍	X
EML(MB2147-05)	X	❍	X
MON	❍	X	X

-From Initialize

Indicates the cumulative total value of execution after the [Clear] button has been clicked.

-From Last Executed

Indicates the immediately preceding execution time.

Note:

The measurement results have errors. For details, refer to "Measuring Execution Time" in each chapter of SOFTUNE WORKBENCH User's Manual.

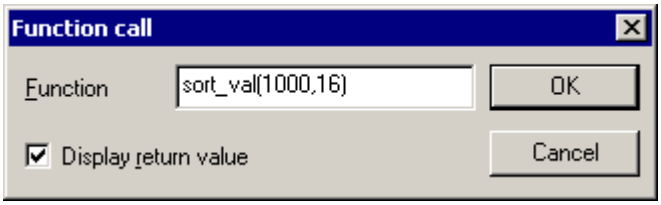
4.6.9 Call

This section explains the SOFTUNE WORKBENCH function call function.

■ Function call

The specified function can be started during debugging without reference to the flow of the program. This function is known as function call.

Figure 4.6-24 Function Call Dialog Box



When the function call dialog box shown in Figure 4.6-24 opens, specify the function you want to call with a correct argument.

Compiles with C arguments, however, next argument does not specify. If you specify it, debugger outputs error message.

- structure
- union
- class

Figure 4.6-25 Function Call Result Dialog Box

When processing of the called function is terminated and control is returned, the function call result dialog box shown in Figure 4.6-25 opens.

[Example]

When function definition is int sub (int param);, specify the function call as follows:

sub (10): When a constant value is directly specified

sub (ii): When variable ii is directly specified

■ Description

Execute the specified functions to display the return values. The return value is set to the built-in variable %CALL.

Evaluate the argument of the specified function in dummy argument format, and execute it.

If the count of specified actual arguments is more than the count of dummy arguments, evaluate as many counts of actual arguments as that exceeds the dummy arguments in an int type.

When the program hits a break point while the CALL command is executing a function, a break occurs at that position.

To execute the call command continuously, use the GO command.

To terminate the execution of the CALL command, use the CLEAR CALL command.

Note that it is not possible to nest this command.

The CALL command sets the break point for the current PC, and sets the return address so as to return to that point, calling a function.

Therefore, if the function called by the CALL command passes through the current PC, a break occurs in the middle of executing the function.

In such cases, the following message is displayed.

Break at address by Invalid call termination

Break at address by Invalid call termination

Re-starts execution using the GO command to continue the execution of the CALL command.

Note:

The CALL function may change the resources such as the registers, memory or I/O from the state prior to the function call. To restore registers, hold contents prior to the function call and execute the functions, or use CLEAR CALL function.
Other resources are not restored.

4.6.10 Clear Call

This section explains the SOFTUNE WORKBENCH call clear function.

■ Clear Call

"Clear Call" is used to restore the original state without executing the function call (see Section "4.6.9 Call") to the end. This function is used after program execution has been stopped by "Breakpoints...", etc.

When "Clear Call" is executed, control returns from the immediately called function. In this case, the function call result is not displayed because the called function is not executed to the end.

4.6.11 Vector

This section explains how to display and modify SOFTUNE WORKBENCH vectors.

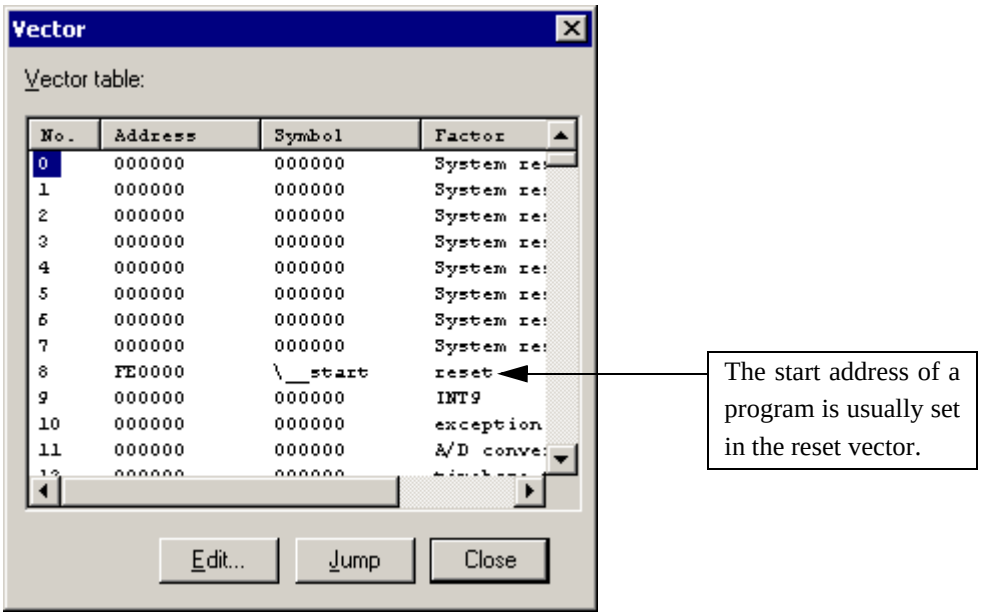
■ Vector

When the MCU is reset or when an interrupt processing request is issued for a variety of factors, the MCU sets the data, set in the address determined in advance according to the type of the interrupt, in the PC as the address of the interrupt processing routine. The address at which this interrupt processing routine is set is called a vector. Vectors are determined in advance according to the kind of the MCU.

■ Display and setting vectors

-Display

Figure 4.6-26 Vector Display Dialog Box



■ Jump

Display the source of the stored program at the address set in the vector table in the following procedure:

1.Select a vector number.

2.Click the [Jump] button.

If the starting address of the program set in the vector table is incorrect, the source cannot be displayed (disassemble display).

Note:

The jump function merely displays the jump destination program; it does not update the program counter to move control to the address set in the vector table.

4.6.12 Load Target File

This section explains how to load the target file to be debugged by SOFTUNE WORKBENCH.

■ Target file

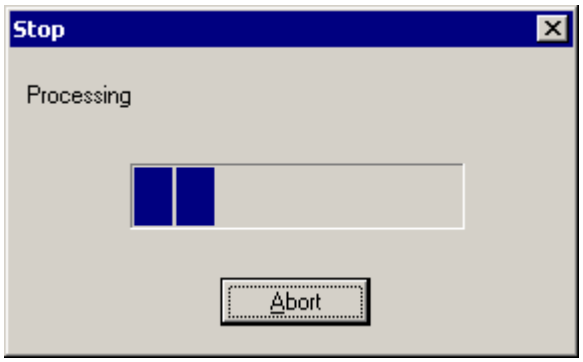
An ABS format target file is to be debugged. This file is registered as a project target file.

Debugging can be started after the ABS format target file has been created. Use SOFTUNE WORKBENCH to create a source program and execute compile/assemble and link. Creation of the ABS format target file is enabled when the program is free from compile/assemble and link errors.

■ Loading the target file

Before loading the target file, select [Start debug] from the [Debug] Menu to place SOFTUNE WORKBENCH in the debug session. When SOFTUNE WORKBENCH enters the debug session, select [Load target file] from the [Debug] Menu to load the target file. The target file load state displays dialog box shown in Figure 4.6-27 opens when the target file is being loaded.

Figure 4.6-27 Target File Load State Display Dialog Box



When loading the target file terminates, an entry point is set in the PC, the source line of the module including the entry point is displayed, and the program stops.

Execute [Step] and [Go], etc., subsequently to continue debugging.

4.6.13 Start Debug/Terminate Debug

This section explains how to start and terminate debugging.

■ Starting debugging

"Start debug" places SOFTUNE WORKBENCH in the debug session to enable the subsequent use of debugger commands. When SOFTUNE WORKBENCH enters the debug session, at first load the target file (see Section "4.6.12 Load Target File").

■ Terminating debugging

"Terminate debug" terminates the SOFTUNE WORKBENCH debug session.

4.7 Setup

"Setup" sets SOFTUNE WORKBENCH execution environment, debugger mode environment, and other tools.

■ Setting the SOFTUNE WORKBENCH execution environment

- Development

■ Setting the debugger mode environment

- Debug Environment
- Memory Map...

■ Setting other tools

- Tool
- Keyboard
- Editor
- Error
- Tool Startup

4.7.1 Development

"Development..." sets SOFTUNE WORKBENCH operation and the environment variables required by language tools (e.g., compiler).

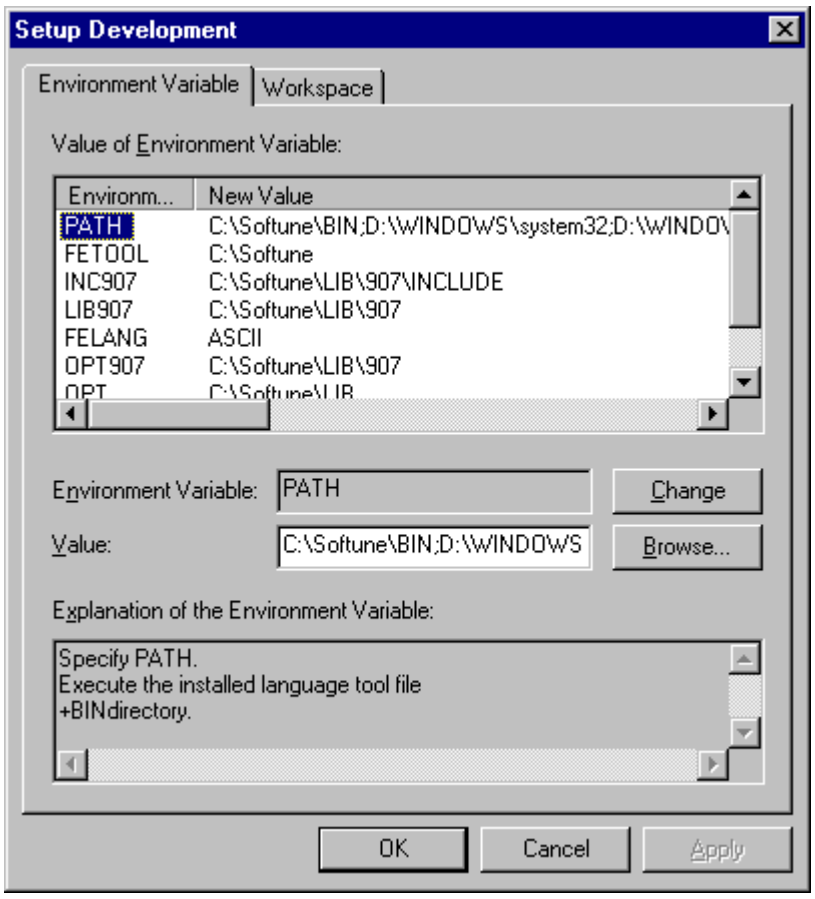
■ Environment variable

Figure 4.7-1 shows the environment variable setup dialog box. The environment variables listed in Table 4.7-1 are set in this section.

Table 4.7-1 Environment Variable Names

Environment variable name	Explanation
FETOOL	Standard directory in which language tools were installed
INC907	Directory in which include file exists
LIB907	Directory in which library file exists
OPT907	Directory in which the language tool default option file exists
FELANG	Character code system switching in the messages output by language tools (SJIS: Japanese language (shift JIS), ASCII: English)
PATH	Directory in which language tools exist
TMP	Directory in which work files exist

Figure 4.7-1 Development Environment Setup (Environment Variable) Dialog Box



Set the development environment in the following procedure:

1.Select the name of the environment variable whose setting is to be changed from the [Value of Environment Variable] list.

The current setting value is displayed in the [Value] field.

Simple explanation of the environment variable is displayed in [Explanation of the Environment Variable].

1.Change the description of the [Value] field.

2.Click the [Change] button.

■ Workspace

"Workspace" sets the following SOFTUNE WORKBENCH operations:

-Open the last workspace at starting

Setting a check mark to the left of this item enables the opening of the previously opened workspace file when SOFTUNE WORKBENCH is started.

-Output tool option at compile/assemble

Setting a check mark to the left of this item enables the display of the options, specified when the C compiler or assembler is started, in the Output Window.

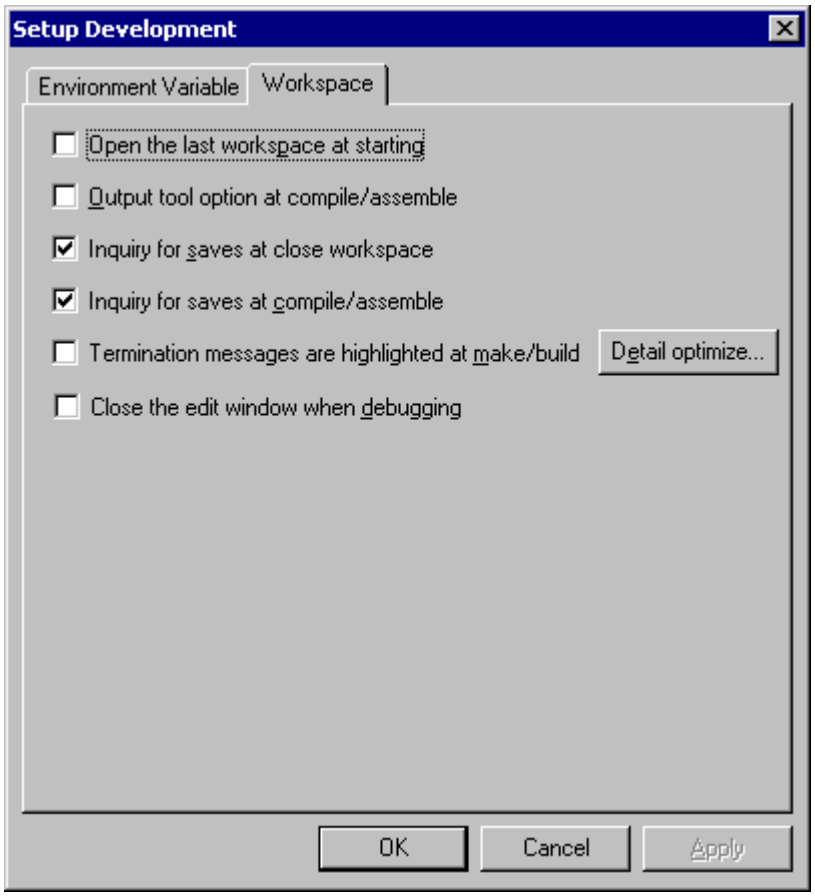
-Inquiry for save at close workspace

When a check mark is set to the left of this item, SOFTUNE WORKBENCH asks you whether to save the currently open file to the workspace file when the workspace is closed.

-Inquiry for save at compile/assemble

When a check mark is set to the left of this item, SOFTUNE WORKBENCH asks you whether to save the file currently being edited before compile/assemble.

Figure 4.7-2 Development Environment Setup (Workspace) Dialog Box

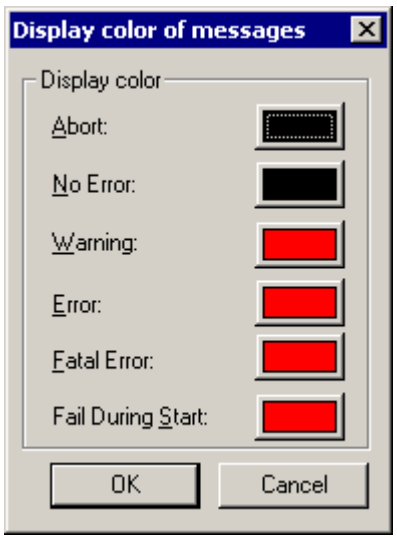


-Termination messages are highlighted at make/build

When this item is checked, the display color of termination messages (abort, no error, warning, error, fatal error, or fail during start) during Compile, Assemble, Make, or Build, can be changed.

To change the display color, click the [Detailed optimize] button at the right of this field; the termination message display color dialog is displayed (Figure 4.7-3). Change the display color.

Figure 4.7-3 Display Color of Messages Dialog Box



-Close the edit window when debugging

By checking this item on a box, edit window, which opens as debugger is started, will be closed.

4.7.2 Debug Environment

"Debug Environment" sets the debug environment; it is valid only when SOFTUNE WORKBENCH is in the debug session.

■ Items to be set

Select and set the following items from the submenu:

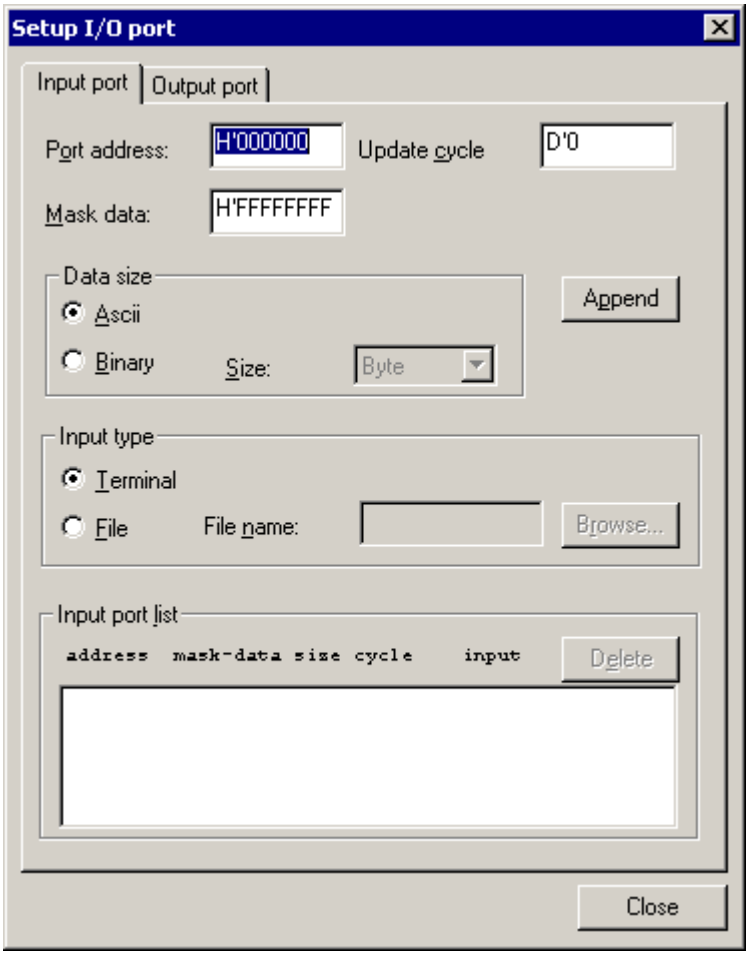
- I/O Port
- Interrupt
- Debug Environment
- Selecting Debug Function
- Setup Wizard

4.7.2.1 I/O Port

This section explains the I/O port setup procedure.

■ Setting an input port

Figure 4.7-4 Input Port Setup Dialog Box



-Port Address

Specifies a port address.

-Mask Data

Specifies address mask. Only the addresses whose bits are 1 are to be compared.

-Data Size

Specifies a data input type. When Binary is selected, specify its size. (Byte/Word/Long)

-Input Type

Specifies a port data input source.

-Input Port List

Displays the currently specified ports.

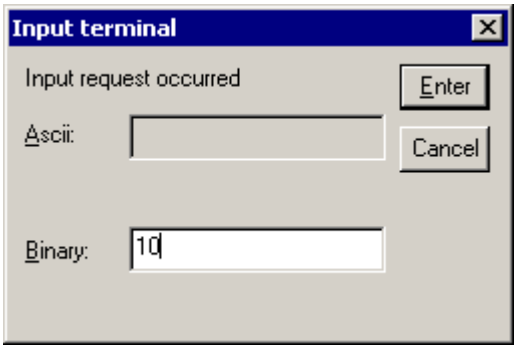
-Input terminal

When an input request is issued during program execution with [Input Type] set to [Terminal], the input terminal dialog box opens.

Specifying [ASCII] as [Data Type] in input port setting enables ASCII input.

Specifying [Binary] as [Data Type] enables binary input.

Figure 4.7-5 Input Terminal Dialog Box

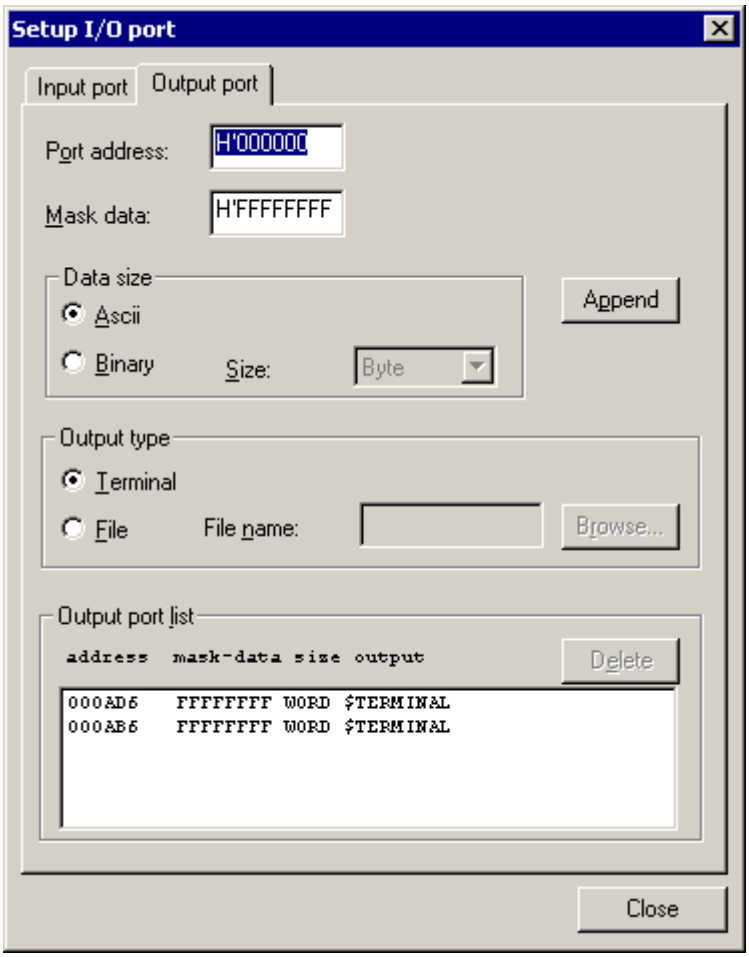


■ Resetting an input port

1. Select the input ports to be reset from [Input Port List].
2. Click the [Delete] button.
3. When resetting all the selected ports is completed, click the [Close] button.

■ Setting an output port

Figure 4.7-6 Output Port Setup Dialog Box



-Port Address

Specifies a port address.

-Mask Data

Specifies address mask. Only the addresses whose bits are 1 are to be compared.

-Data Size

Specifies a data output type. When Binary is selected, specify its size. (Byte/Word/Long).

-Output Type

Specifies a port data output destination.

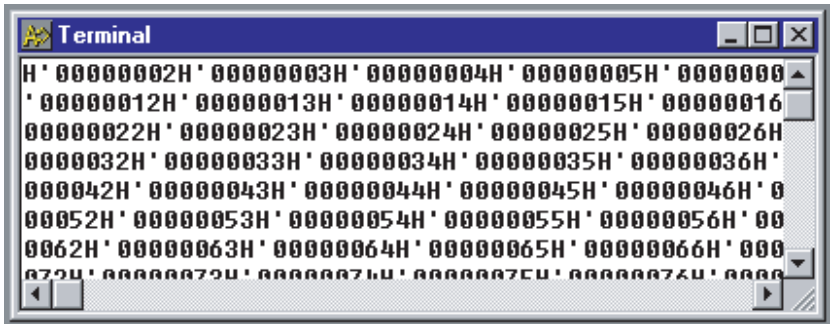
-Output Port List

Displays the currently specified ports.

-Output terminal

When an output request is issued during program execution with [Output Type] set to [Terminal], the Terminal Window is displayed. The output type also depends on [Data Type].

Figure 4.7-7 Output Terminal Window (Binary)



■ Resetting an output port

1. Select the output ports you want to reset from [Output Port List].
2. Click the [Delete] button.
3. When resetting all the selected ports is completed, click the [Close] button.

4.7.2.2 Interrupt

This section explains the interrupt setup procedure.

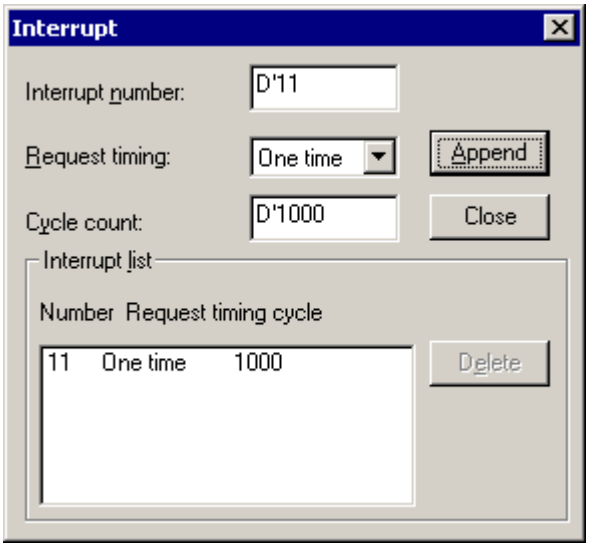
■ Setting an interrupt

1. When the interrupt setup dialog box shown in Figure 4.7-8 opens, set an interrupt number.
2. Select an issuance timing.
   [One Time] or [Interval] can be selected.
3. Set the interrupt cycle count.
4. Click the [Add] button.
   The set interrupt number, issuance timing, and interrupt cycle count are displayed in [Interrupt List].
5. When setting all the items is completed, click the [Close] button.

■ Resetting an interrupt

1. When the interrupt setup dialog box shown in Figure 4.7-8 opens, set an interrupt number.
2. Select the interrupt to be reset from [Interrupt List].
3. Click the [Delete] button.
4. When resetting all the selected interrupts is completed, click the [Close] button.

Figure 4.7-8 Interrupt Setup Dialog Box



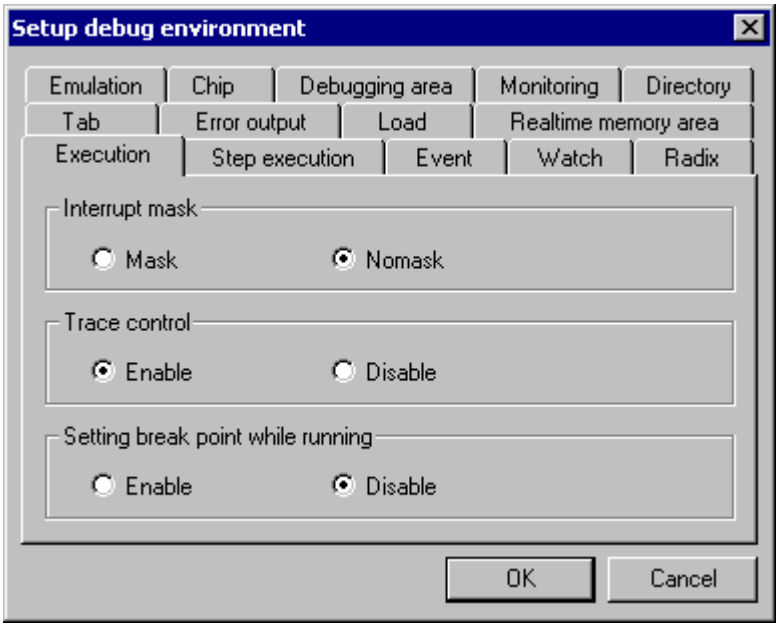
4.7.2.3 Debug Environment

This section explains the debug environment setup procedure.

■ Debug environment setup procedure

-Execution

Figure 4.7-9 Debug Environment Setup Dialog Box (Execution)



-Interrupt Mask

Specifies whether to enable/disable interrupt mask during step execution.

-Trace Control

Specifies whether to enable/disable trace control.

-Setting break point while running

If "Setting break point while running" is enabled, it is also possible to Break settings even when executing a user program.

-Step Execution

Figure 4.7-10 Debug Environment Setup Dialog Box (Step Execution)

-Step Scale

-Auto

Automatically sets the step unit according to the window display state.

-Source Line

Executes the step in units of source lines.

-Instruct

Executes the step in units of machine languages.

-Interrupt Mask

Specifies whether to enable/disable interrupts.

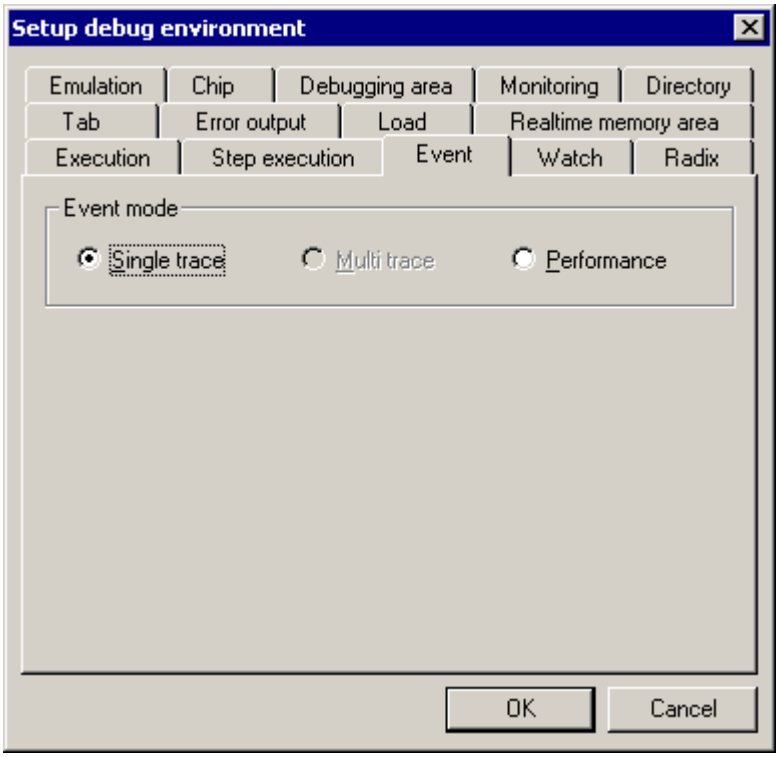
-System Call Debug

Specifies skip operation for the INTP instruction to be executed during step-in operation.

-Event [MB2141/MB2147-01]

Note: Event is valid only at emulator debugger.

Figure 4.7-11 Debug Environment Setup Dialog Box (Event)



-Event Mode

Specifies an event mode.

-Single Trace

Specifies the normal mode.

-Multi-Trace

Specifies the multi-trace mode.

-Performance

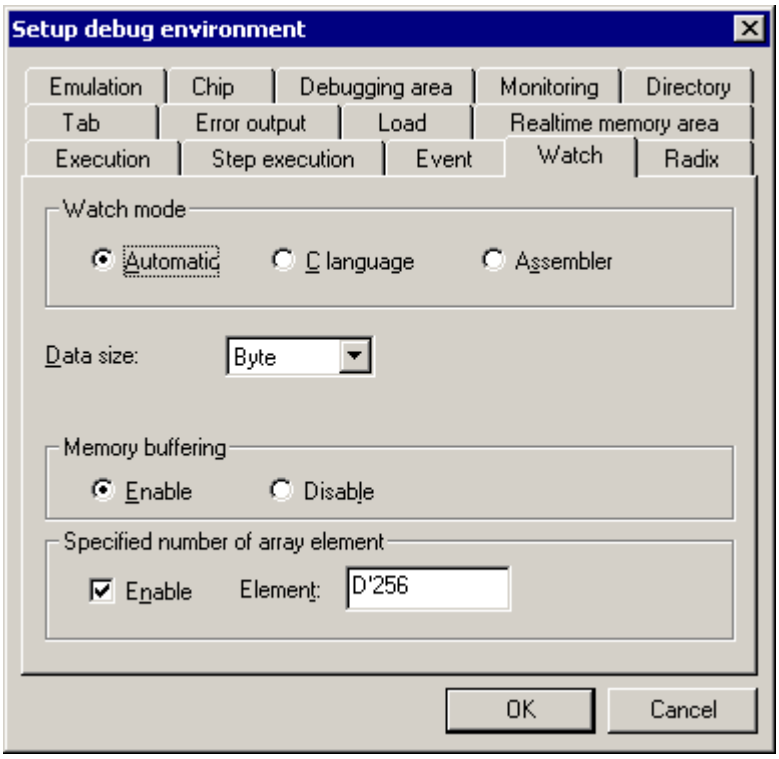
Specifies the performance mode.

Note:

Multi trace is enabled only for the MB2147-01 debug function in the Trace Enhancement mode. For more information, see Section "2.3.1.6 Debug Function" in SOFTUNE Workbench User's Manual.

-Watch

Figure 4.7-12 Debug Environment Setup Dialog Box (Watch)



-Watch Mode

-Automatic

Sets the watch mode automatically according to the analysis result.

-C Language

Sets the C language mode (interpretation as C language expressions).

-Assembler

Sets the assembler mode (interpretation as assembler expressions).

-Data Size

Sets the display size in the assembler mode.

-Byte/word/long/single/double

-Memory Buffering

-Enable

In case of variables as arrays or structures, the memory of whole variables is read.

They are accessed by size of the top variable.

-Disable

In case of variables as arrays or structures, the memory of each element and member unit is read.

-Specified number of array element

-Enable

Debugger displays a warning dialogue in case of big array element larger than the number of limited array-element when you registering or expanding an array with a watch variable.

-Element

You specify number (a default is D'256) of array element.

The default of this control is "Enable".

If "Disable" is selected, the watch window and the local window may be displayed slowly.

Note:

If memory buffering is set as valid, correct value such as I/O to request lead of fixed size can not be displayed.

-Radix

Figure 4.7-13 Debug Environment Setup Dialog Box (Radix)

-Radix

Sets the base number for numerical value display and analysis.

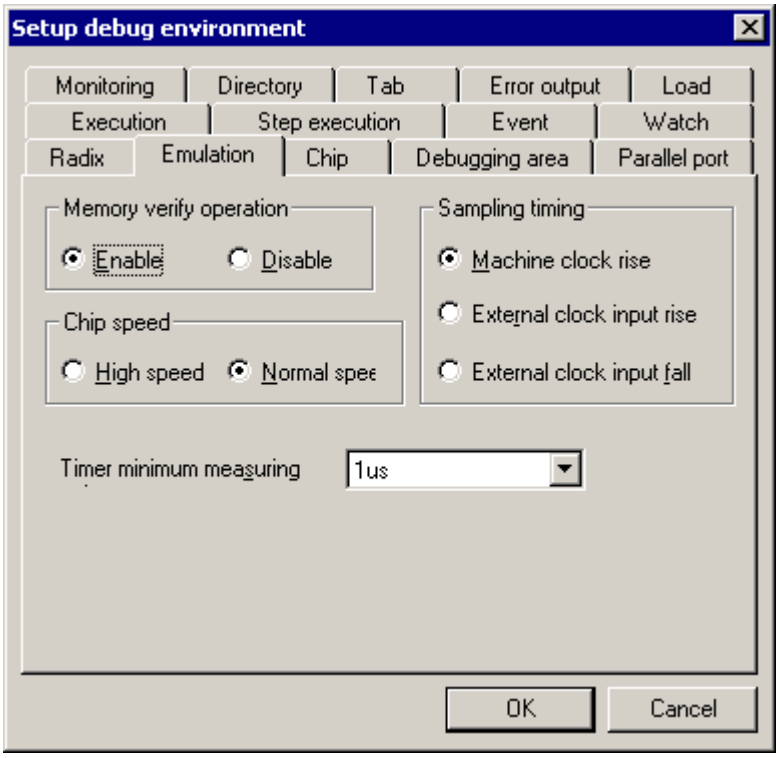
-Display Source Line

Switches source line display and nondisplay.

-Emulation [MB2141]

Note: Emulation is valid only at emulator debugger.

Figure 4.7-14 Debug Environment Setup Dialog Box (Emulation)



-Memory Verify Operation

Specifies whether to verify memory when data is written to memory.

-Chip Speed

Specifies an MCU operation speed (normal/high speed).

-Sampling Timing

-Machine Clock Rising

Sets the rising of the machine clock as sampling timing.

-External Clock Input Rising

Sets the rising of the external clock as sampling timing.

-External Clock Input Falling

Sets the falling of the external clock as sampling timing.

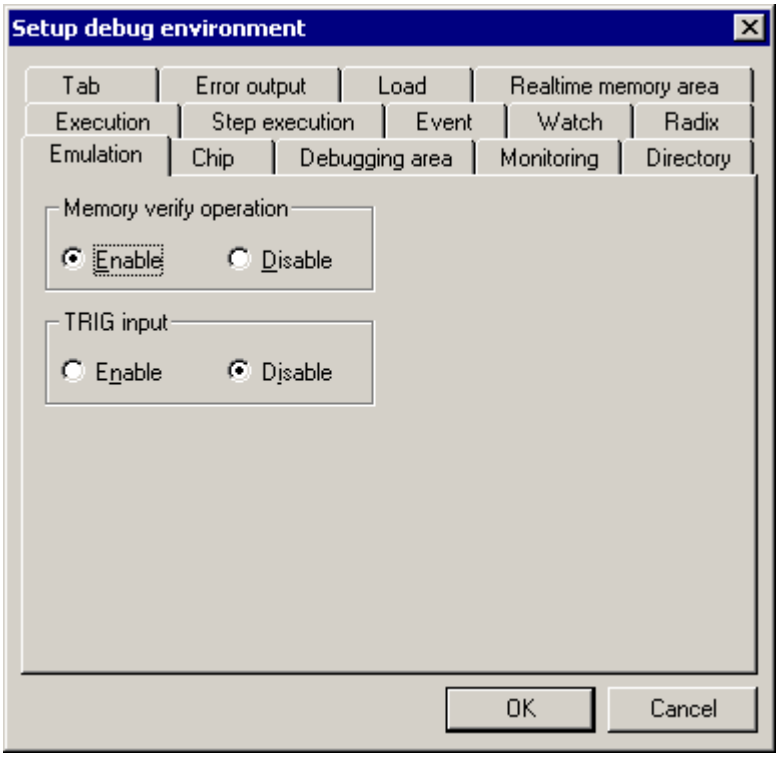
-Timer minimum measuring

Specifies the minimum unit for execution time measurement. (1µs/100ms)

-Emulation [MB2147-01]

Note: Emulation is valid only at emulator debugger.

Figure 4.7-15 Debug Environment Setup Dialog Box (Emulation)



-Memory Verify Operation

Specifies whether to verify memory when data is written to memory.

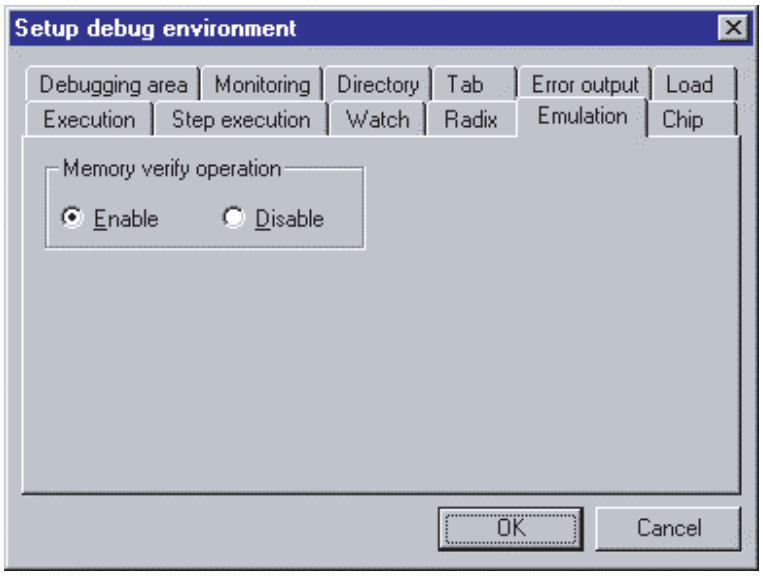
-TRIG Input

Specifies whether to enable or disable TRIG pin input.

-Emulation [MB2147-05]

Note: Emulation is valid only at emulator debugger.

Figure 4.7-16 Debug Environment Setup Dialog Box (Emulation)



-Memory Verify Operation

Specifies whether to verify memory when data is written to memory.

-Chip [MB2141/MB2147-05/MB2147-01]

Note: Chip is valid only at emulator debugger.

Figure 4.7-17 Debug Environment Setup Dialog Box (Chip) [MB2141/MB2147-05]

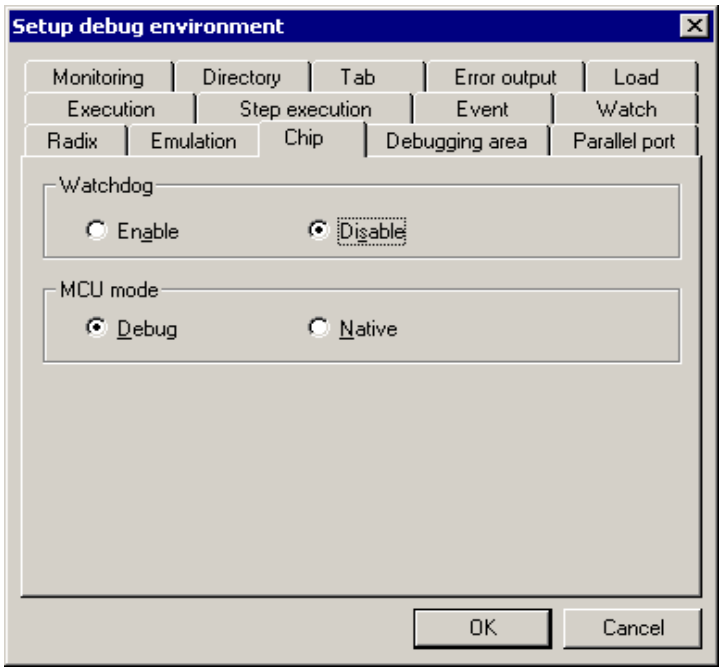
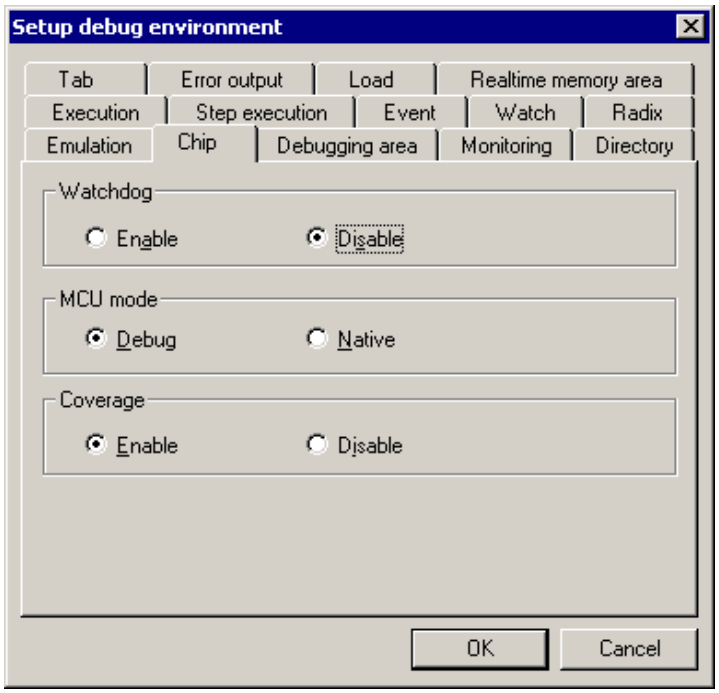


Figure 4.7-18 Debug Environment Setup Dialog Box (Chip) [MB2147-01]



-Watchdog

Specifies whether the watchdog timer is valid or invalid.

-MCU Mode

-Debug

Specifies the debug session mode.

-Native

Specifies the native mode.

-Coverage (Only MB2147-01)

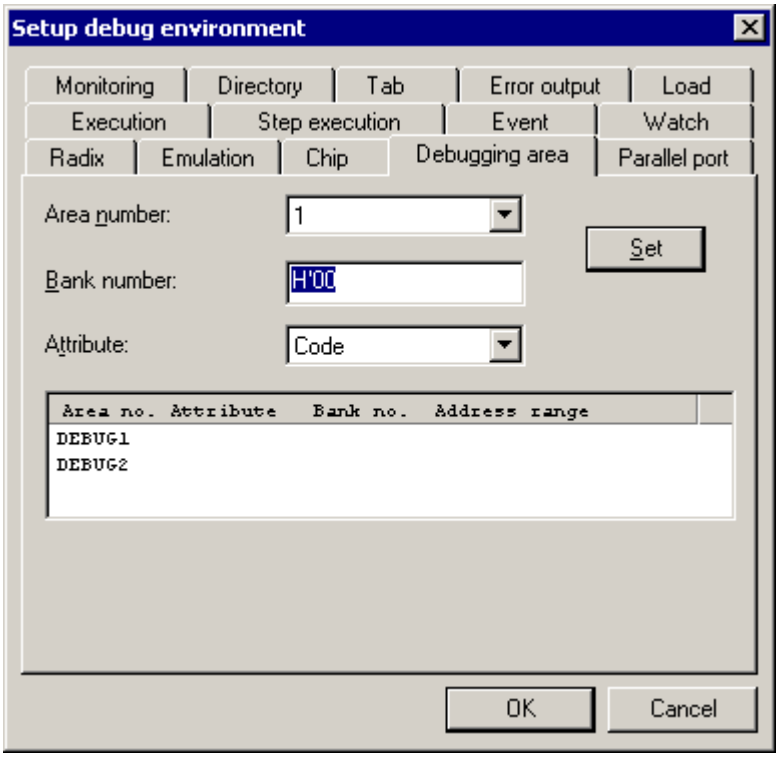
Enable to specify in MB2147-01.

In executing a user program, the memory of the 0 bank (internal I/O and internal RAM) can be referenced by disabling the coverage function.

-Debug Area [MB2141]

Note: Debug Area is valid only at emulator debugger.

Figure 4.7-19 Debug Environment Setup Dialog Box (Debugging Area)



-Area Number (1/2)

Specifies an area number.

-Bank Number

Specifies a bank number.

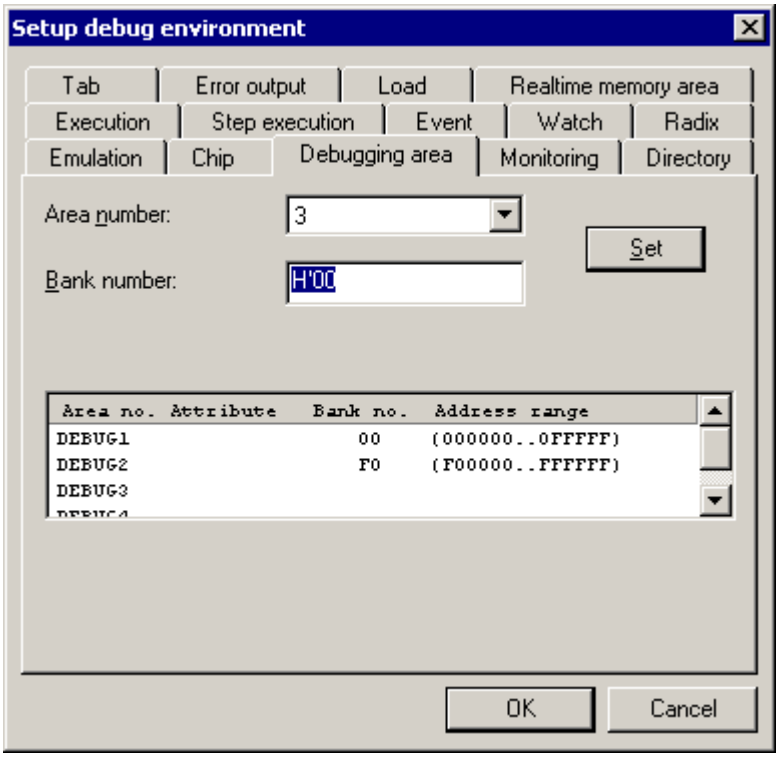
-Attribute

Specifies an attribute (code, read, write, read/write).

-Debugging Area [MB2147-01/MB2147-05]

Note: Debug Area is valid only at emulator debugger.

Figure 4.7-20 Debug Environment Setup Dialog Box (Debugging Area)



-Area Number (1/2)

The area number (3, 4) is specified. Area 1 is fixed at 0x000000 to 0x0fffff. Area 2 is fixed at 0xf00000 to 0xffffff. Neither area can be changed. Also, the same address area cannot be set in another area.

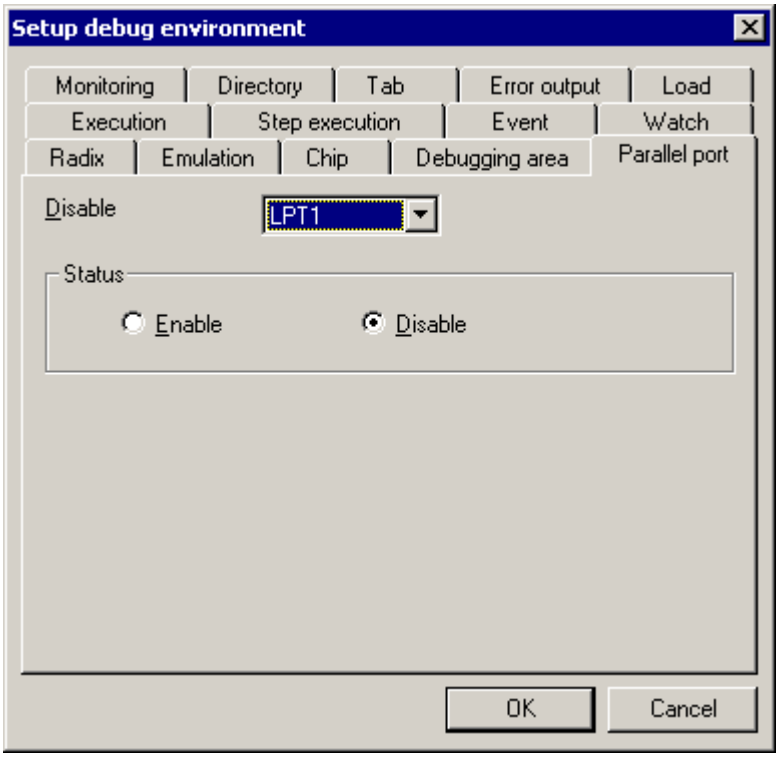
-Bank Number

Specifies a bank number.

-Parallel Port [MB2141]

Note: Parallel Port is valid only at emulator debugger.

Figure 4.7-21 Debug Environment Setup Dialog Box (Parallel Port)



-Port Name

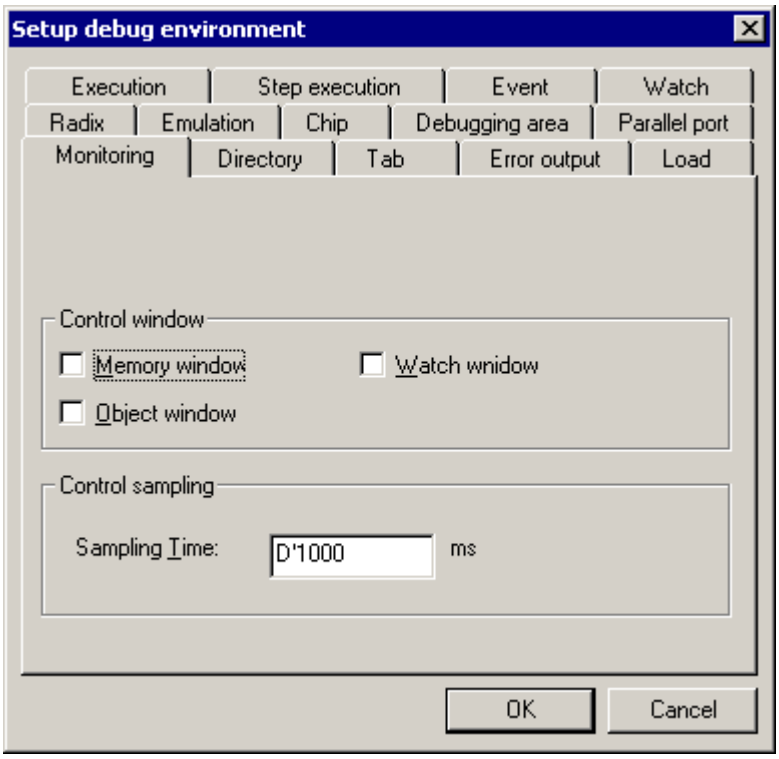
Specifies non-connection or a parallel port name.(None/LPT1/LPT2)

-State

Specifies whether the parallel port is valid or invalid.

-Monitoring[MB2141/MB2147-05/Simulator]

Figure 4.7-22 Debug Environment Setup Dialog Box (Monitoring) [MB2141/MB2147-05/Simulator]



-Monitoring Control

Sets monitoring control.

-Memory Window

Specifies whether to monitor the Memory Window.

-Watch Window

Specifies whether to monitor the Watch Window.

-Object Window

Specifies whether to monitor the Object Window.

-Sampling Time

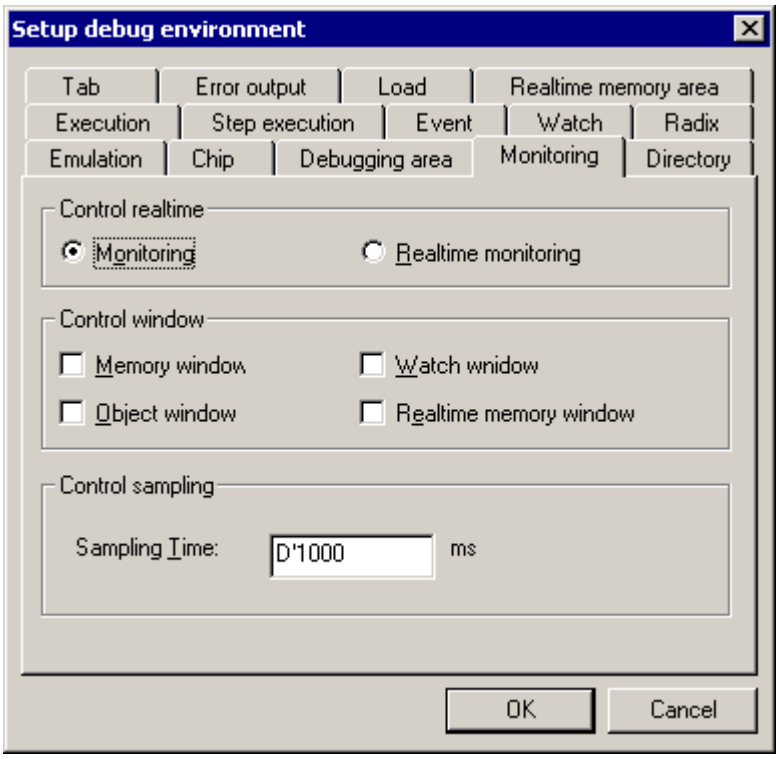
Specifies sampling time.

MB2141 : minimum 1000ms

MB2147-05: minimum 100ms

-Monitoring[MB2147-01]

Figure 4.7-23 Debug Environment Setup Dialog Box (Monitoring) [MB2147-01]



-Monitoring Control

Sets monitoring control.

-Monitoring

On-the-fly monitoring is performed.

-Real-time monitoring

On-the-fly monitoring is inhibited and only the mirror memory is monitored.

When real-time monitoring is selected, the memory window and object window are not monitored; only the real-time area is monitored on the watch window.

-Memory Window

Specifies whether to monitor the Memory Window.

-Watch Window

Specifies whether to monitor the Watch Window.

-Object Window

Specifies whether to monitor the Object Window.

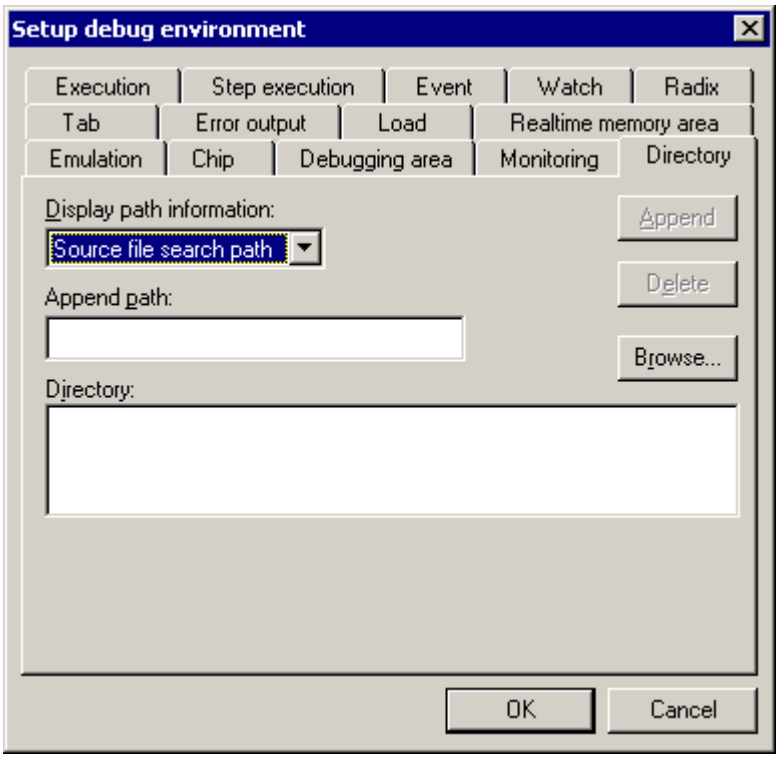
-Sampling Time

Specifies sampling time.

MB2147-01: minimum 100ms

-Directory

Figure 4.7-24 Debug Environment Setup Dialog Box (Directory)



-Display Path Information

Specifies the path information to be displayed.

-Append Path

Sets the path to be added.

-Directory

Displays the currently set directory.

-Directory setup procedure

1.Select the [Display Path Information] to be displayed.

2.Set the [Append Path].
Clicking the [Browse] button to the right of the [Append Path] setup field enables path selection.

3.Click the [Append] button.

4.When there is no other item to be set, click the [OK] button.

-Directory reset procedure

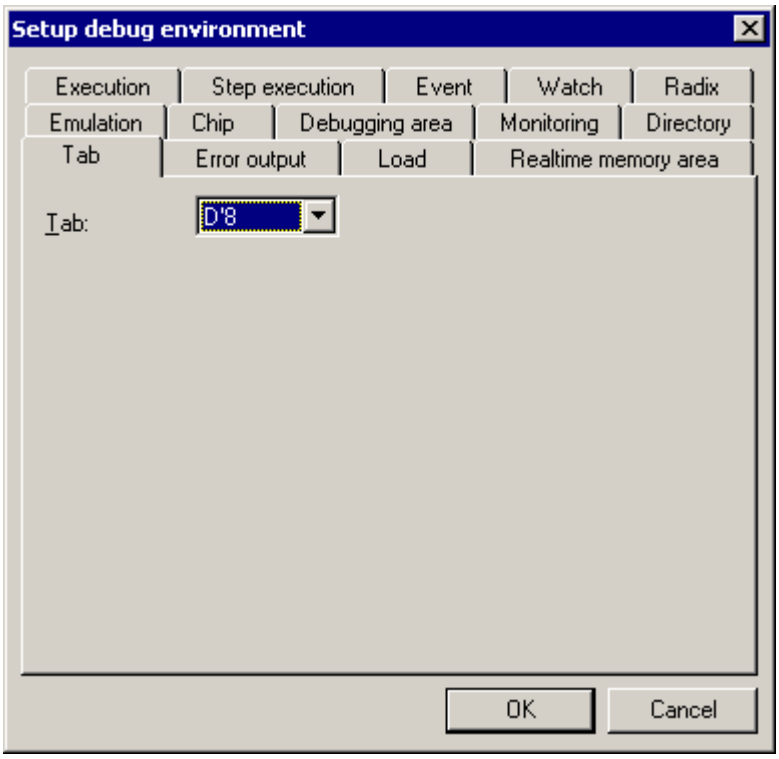
1.Select the directory you want to delete from [Directory].

2.Click the [Delete] button.

3.When there is no other item to be set, click the [OK] button.

-Tab

Figure 4.7-25 Debug Environment Setup Dialog Box (Tab)

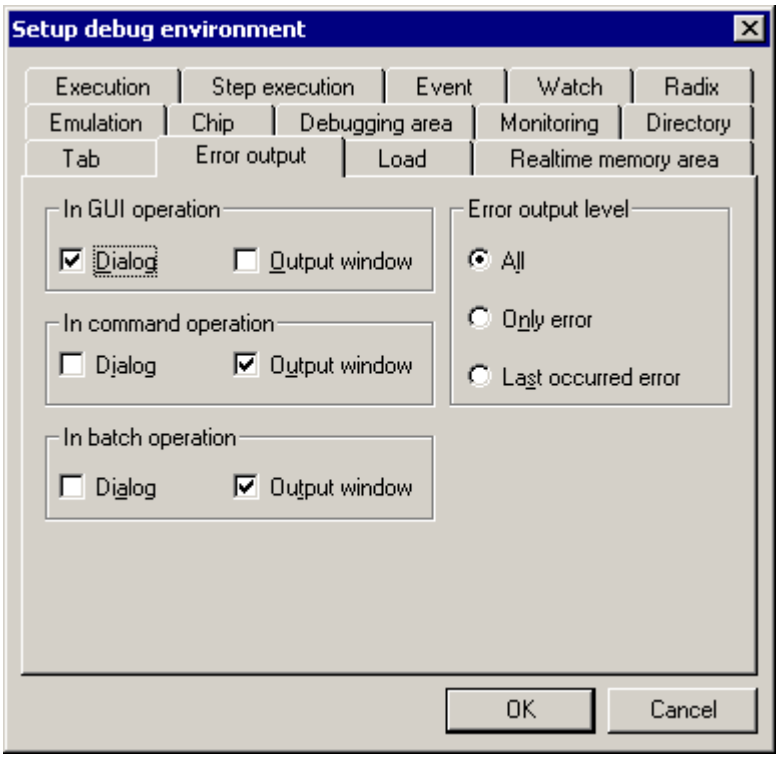


-Tab

Specifies the Tab.(D'4/D'8)

-Error output

Figure 4.7-26 Debug Environment Setup Dialog Box (Error Output)



-In GUI Operation

Specifies an error output type at GUI operation.

-In Command Operation

Specifies an error output type at command operation.

-In Batch Operation

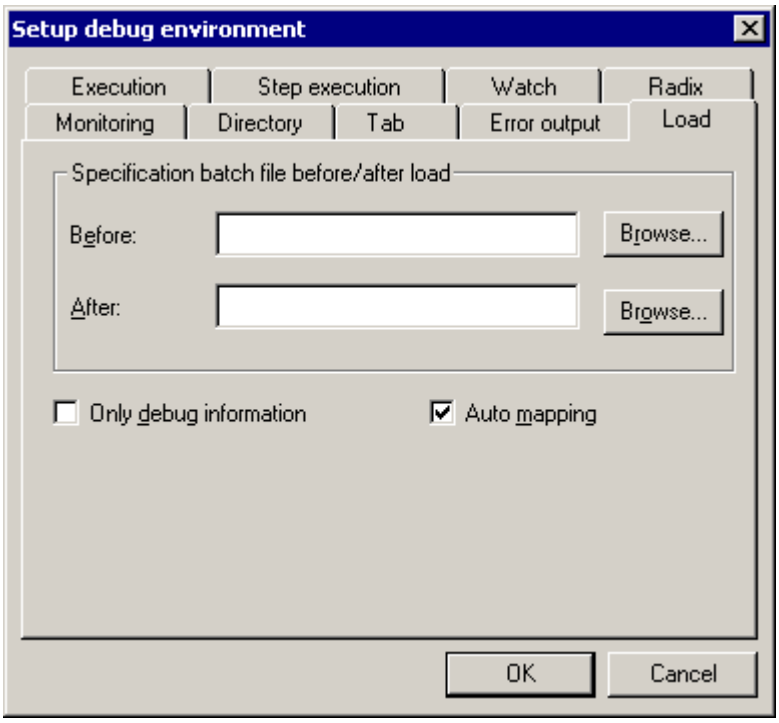
Specifies an error output type at batch operation.

-Error Output Level

Sets the output type when several errors occur.

-Load

Figure 4.7-27 Debug Environment Setup Dialog Box (Load)



This sets the environment when loading a target file registered in the project.

-Pre-Post Load Batch File Specification

Pre- Load

This specifies the batch file to execute prior to the loading of the target file. This can be changed using the Debugger's setup wizard.

Post- Load

This specifies the batch file to execute after the loading of the target file. This can be changed using the debugger's setup wizard.

-Debug Information Only

This specifies whether or not to load only the debug information or not. When checked, only the debug information is loaded.

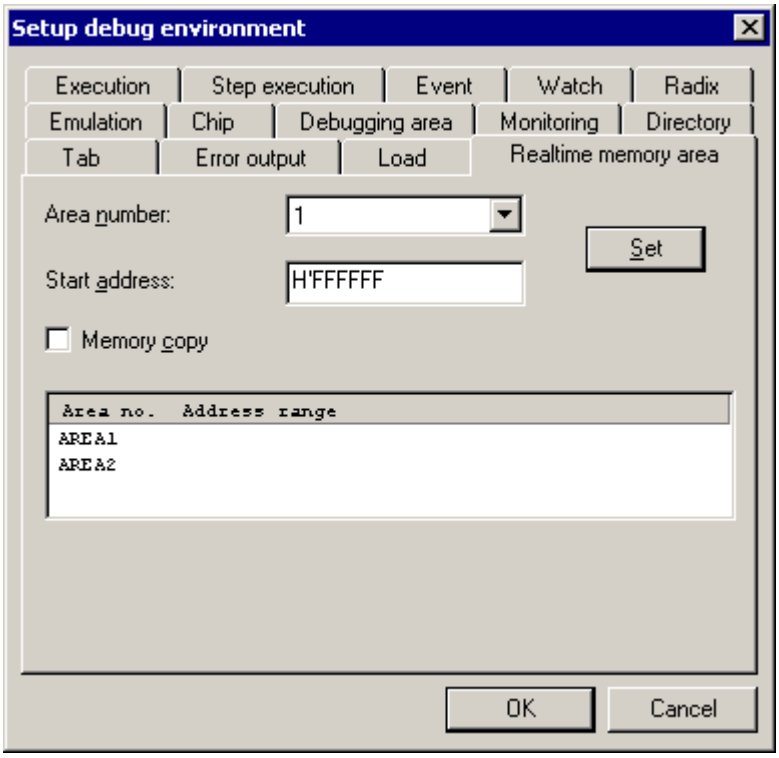
-Auto-Map Setting

This specifies whether or not to enable the Auto-Map Setting. When checked, Auto-Map Setting is enabled.

-Real-time memory area [MB2147-01]

Note: Real-time memory area is valid only at emulator debugger.

Figure 4.7-28 Debug Environment Setup Dialog Box (Realtime memory Area)



-Area Number

Specifies an area number.(1/2)

-Starting address

The starting address of the real-time area is specified. The lower 8 bits are masked.

-Memory copy

The contents of real memory are displayed at opening of the window.

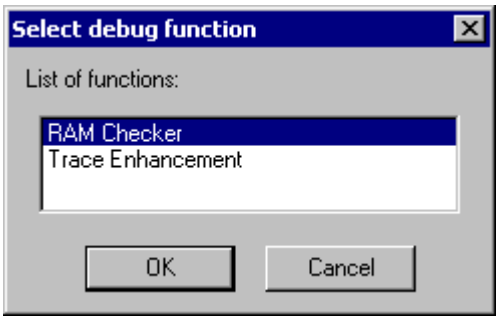
4.7.2.4 Selection Debug Function

This section explains the procedure to select the debug function.

■ Procedure to Select Debug Function

1. Choose functions you want to use from [List of functions].
2. Click [OK] button or double click the function name.
   Debug function is switched to the one you selected.

Figure 4.7-29 Select Debug Function



Note:

The available debug functions vary depending on the emulator or its connecting form. If there is no function available, [Select debug function] menu is disabled.
For each debug function, see Section "2.3.1.6 Debug Function" in SOFTUNE Workbench User's Manual.
When you change the debug function, the data in trace and performance will be cleared. RAM Checker mode is set when activated.

4.7.2.5 Setup Wizard

This section explains how to operate the debugger's Setup Wizard.

■ Setup Wizard operation procedure

1. Select [Setup Name] from [Debugger Startup Selection], then click the [OK] button.
   The check dialog box opens.
2. Click the [OK] button.
   Setup Wizard is started.
3. Click the [Next] button.
4. Select a debugger type, then click the [Next] button.
   Setting of the subsequent items depends on the debugger’s type.
   To reset an item, click the [Return] button.
   The immediately preceding setup screen is redisplayed.

■ Procedure when the emulator debugger is selected

1. Select the emulator type. Select [RS232C], [LAN] or [USB] as the device type.
   When [RS232C] is selected, set a port and band rate.
   When [LAN] is selected, set a host name.
2. Click the [Next] button.
3. Select [RS232C] or [LAN] or [USB] as the device type.
   When [RS232C] is selected, set a port and baud rate.
   When [LAN] is selected, set a host name.
4. Click the [Next] button.
5. Set whether or not to automatically load the monitor program at debugging start.
6. Click the [Next] button.
7. Set whether or not to load the target file automatically during debugging.
8. Specify the batch files used before and after load as required.
   Clicking the [Refer] button to the right of each input field enables batch file selection.
9. Click the [Next] button.
10. Select the item to be set in the setup file.
    When [Specify] is selected, click the [Setup] button to open the item selection dialog box, select the item to be set from the dialog box, then click the [OK] button.
11. Click the [Next] button.
12. Click the [Complete] button.
    Emulator debugger setup has been just completed.

■ Procedure when the simulator debugger is selected

1. Set whether to load the target file automatically during debugging.
2. Specify the batch files used before and after load as required.
   Clicking the [Refer] button to the right of each input field enables batch file selection.
3. Click the [Next] button.
4. Select the item to be set in the setup file.
   When [Specify] is selected, click the [Setup] button to open the item selection dialog box, and select the item to be set from the dialog box, then click the [OK] button.
5. Click the [Next] button.
6. Click the [Complete] button.
   Simulator debugger setup has been just completed.

■ Procedure when the monitor debugger is selected

1. Select [RS232C] as the device type.
   When [RS232C] is selected, set a port and baud rate.
2. Click the [Next] button.
3. Set whether or not to load the target file automatically during debugging.
4. Specify the batch files used before and after load as required.
   Clicking the [Refer] button to the right of each input field enables batch file selection.
5. Click the [Next] button.
6. Select the item to be set in the setup file.
   When [Specify] is selected, click the [Setup] button to open the item selection dialog box, and select the item to be set from the dialog box, then click the [OK] button.
7. Click the [Next] button.
8. Click the [Complete] button.
   Monitor debugger setup has been just completed.

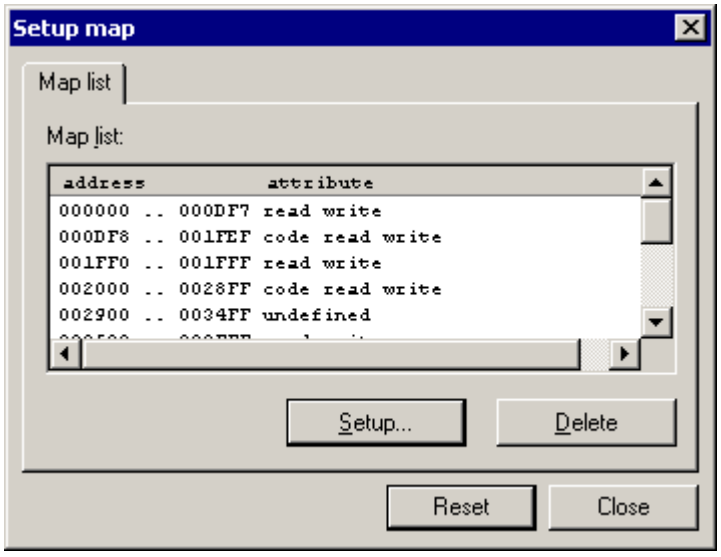
4.7.3 Memory Map

"Memory Map..." sets the debugger's memory map.

■ Memory map setup [SIM]

-Map List

Figure 4.7-30 Memory Map Setup Dialog Box



-Map Area

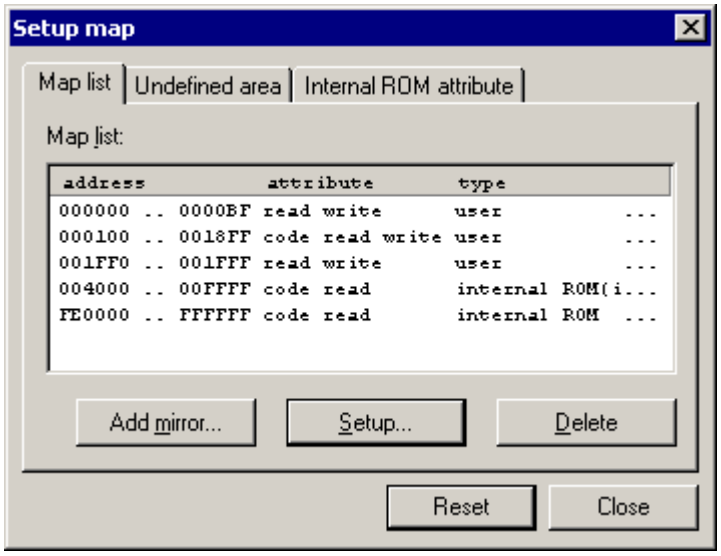
Displays the currently set map area.

-Clicking the [Setup] button opens the setup dialog box (Figure 4.7-30) corresponding to the debug session.

■ Memory map setup [MB2141 EML]

-Map List

Figure 4.7-31 Memory Map Setup Dialog Box



-Map Area

Displays the currently set map area.

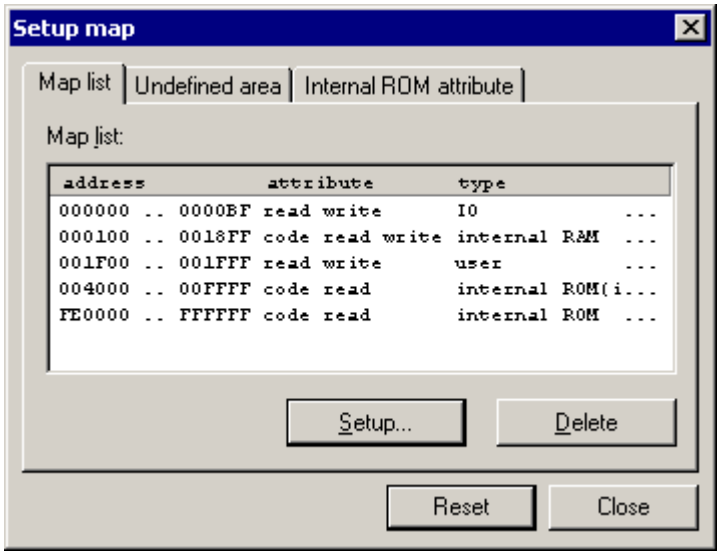
-Clicking the [Add Mirror] button opens the mirror area setup dialog box (Figure 4.7-33).

-Clicking the [Setup] button opens the setup dialog box (Figure 4.7-34) corresponding to the debug session.

■ Memory map setup [MB2147-01/MB2147-05 EML]

-Map List

Figure 4.7-32 Memory Map Setup Dialog Box



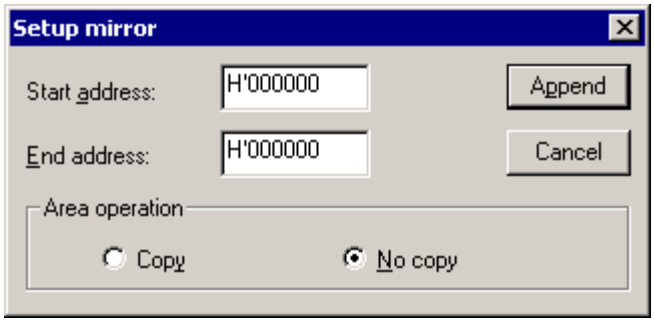
-Map Area

Displays the currently setup map area.

-Clicking the [Setup] button opens the setup dialog box (Figure 4.7-34)

-Mirror Area Setup [MB2141]

Figure 4.7-33 Memory Map Setup Dialog Box



-Start Address

Specifies the start address to be set.

-End Address

Specifies the end address to be set.

-Area Operation

Specifies an area operation type (Copy or Not Copy).

-Map Setup [EML]

Figure 4.7-34 Map Setup Dialog Box [EML]

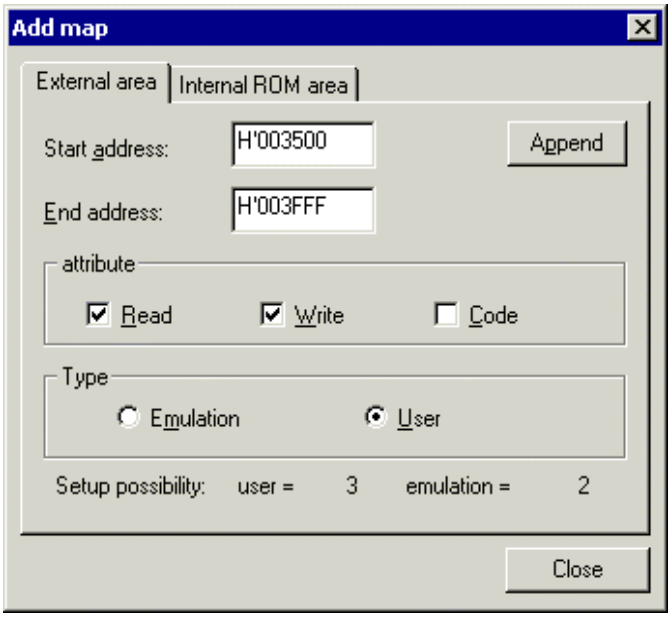
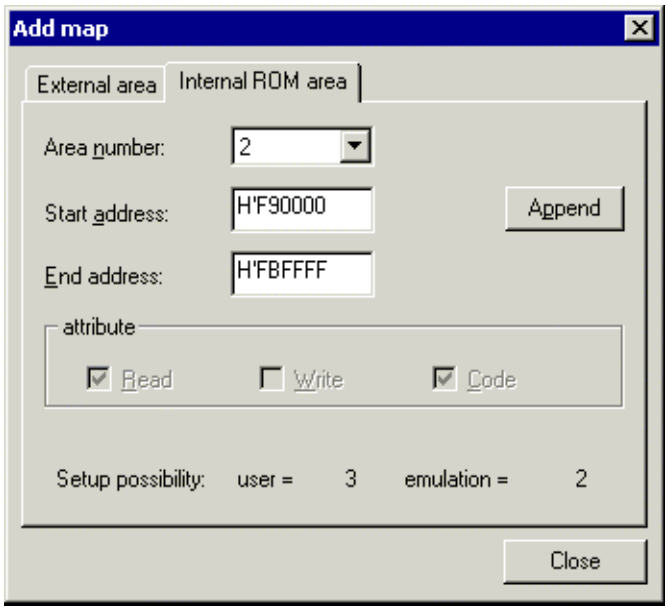


Figure 4.7-35 Map Setup Dialog Box [EML]



-Start Address

Specifies the start address to be set.

-End Address

Specifies the end address to be set.

-Attribute

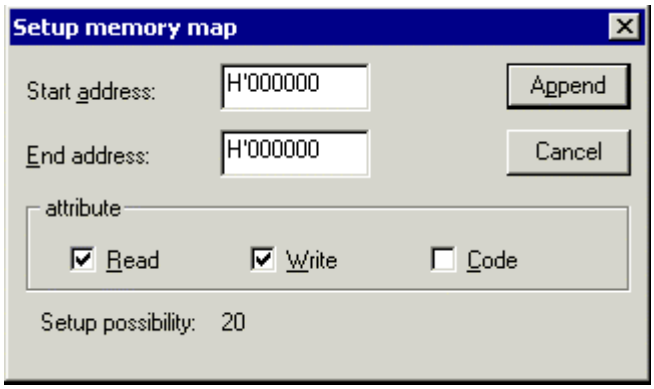
Specifies a memory space attribute (Read, Write, or Code).

-Type

Specifies a setup area type.

-Map Setup [SIM]

Figure 4.7-36 Map Setup Dialog Box [SIM]



-Start Address

Specifies the start address to be set.

-End Address

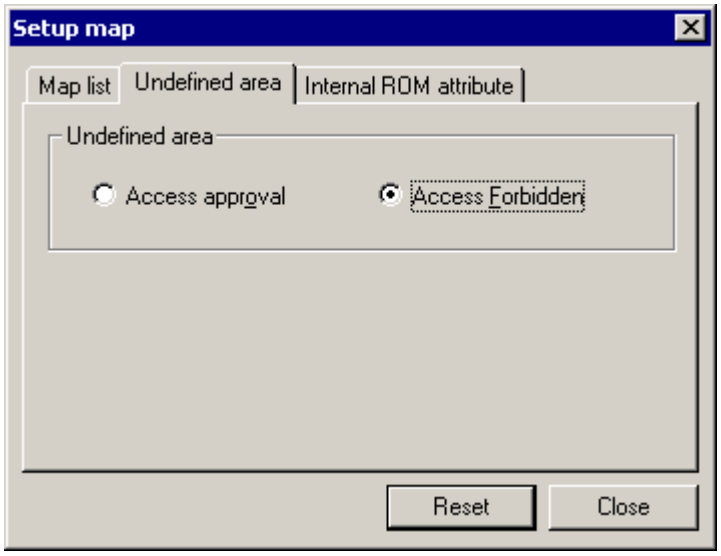
Specifies the end address to be set.

-Attribute

Specifies a memory space attribute (Read, Write, or Code).

-Undefined Area [EML]

Figure 4.7-37 Undefined Area Setup Dialog Box



-Undefined Area

-Access Approval

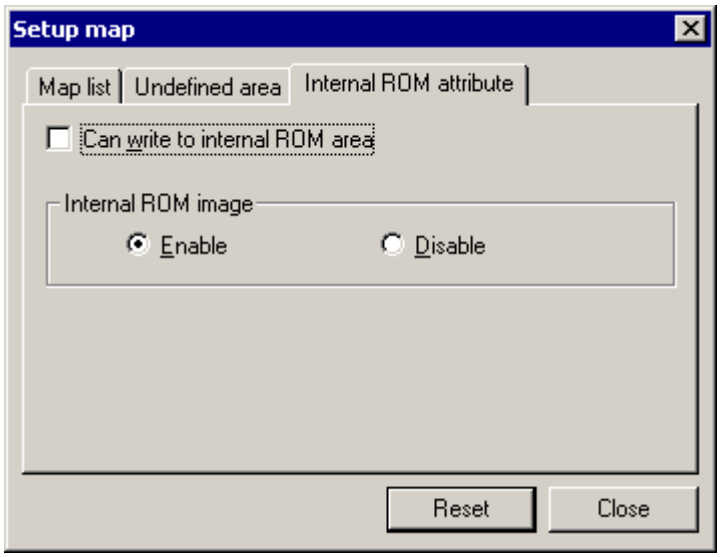
Permits access to the undefined area.

-Access Forbidden

Inhibits access to the undefined area.

-Internal ROM Attribute [EML]

Figure 4.7-38 Internal ROM Attribute Setup Dialog Box [EML]



-Write to Internal ROM area

Sets an internal ROM area write attribute.

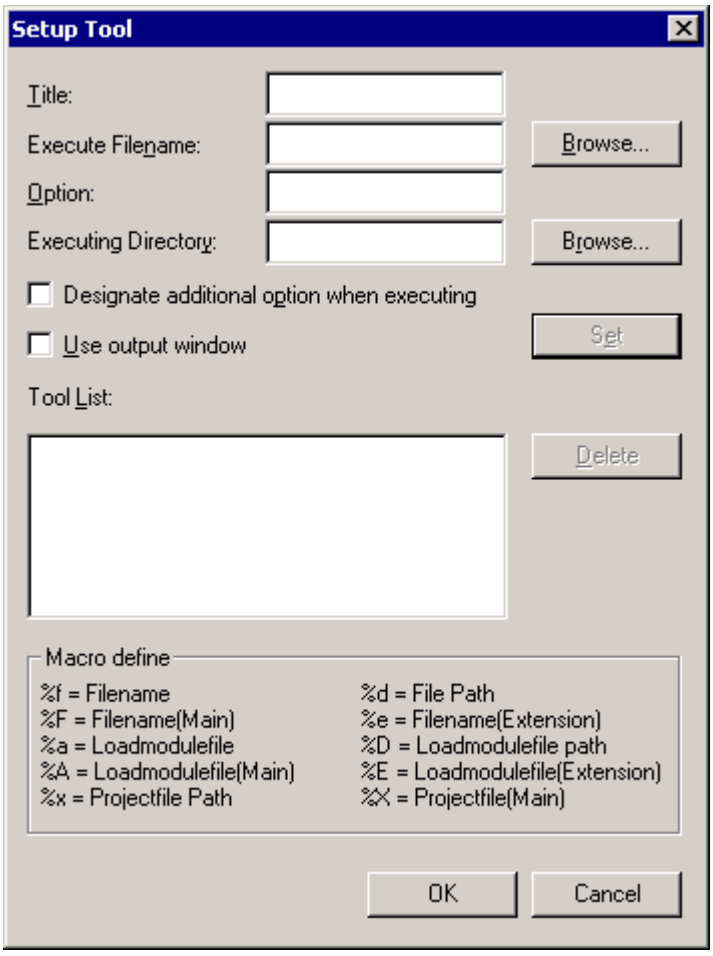
4.7.4 Tool

"Tool..." sets the tools to be directly started by SOFTUNE WORKBENCH.

■ Tools

"Tool..." is not a tool that takes charge of basic SOFTUNE WORKBENCH functions such as a C compiler and assembler. It is a function that builds auxiliary tools (e.g., simple filters) into the system so that they can be started directly from SOFTUNE WORKBENCH. Building "dir" into the system, for example, enables the output of the result obtained as a result of executing the dir command at the DOS prompt to the SOFTUNE WORKBENCH Output Window.

Figure 4.7-39 Tool Setup Dialog Box



■ Tool setup procedure

1. Select [Tool...] from the [Setup] Menu.
   The tool setup dialog box shown in Figure 4.7-39 opens.
2. Set a title that differs from the registered names.
3. Specify the execution file name of the tool to be registered.
   Clicking the [Browse] button to the right of this field opens the file selection dialog box. The execution file name of the tool can be selected from this dialog box.
4. Set an option.
   Macro description can be used in this field. For macro description, refer to Section "1.11 Macro Descriptions Usable in Manager in SOFTUNE WORKBENCH User's Manual".
5. Write an executing directory.
   This description may be omitted if control need not be moved to any specific executing directory.
   Clicking the [Browse] button to the right of this field opens the file selection dialog box. A run-time directory can be selected from this dialog box.
6. Set a check mark to the left of [Designate Additional Option when Executing] and [Use Output Window] as required.
   When a check mark is set to the left of [Designate Additional Option when Executing], SOFTUNE WORKBENCH asks you to enter additional options when a tool is started. When a check mark is set to the left of [Use Output Window], SOFTUNE WORKBENCH displays tool output (output to the standard output device or standard error output device) in the Output Window.
7. Click the [Setup] button.

■ Tool deletion procedure

1. Select [Tool...] from the [Setup] Menu.
   The tool setup dialog box shown in Figure 4.7-39 opens.
2. Select the tool title you want to delete from the tool list.
3. Click the [Delete] button.

■ Tool change procedure

1. Select [Tool...] from the [Setup] Menu.
   The tool setup dialog box shown in Figure 4.7-39 opens.
2. Select the tool title you want to change from the tool list.
   The values set in [Title], [Execute Filename], [Option], [Executing Directory], [Designate Additional Option when Executing], and [Use Output Window] are displayed.
3. Change [Execute Filename], [Option], [Executing Directory], [Designate Additional Option when Executing], and [Use Output Window].
   When [Title] is changed, the set tool is registered as another tool.
4. Click the [Setup] button.
   The dialog box asking you whether to change the tool opens.
5. Click the [Yes] button.

■ Example of tool setup

-When notepad is used

Title	:	Memo pad
Execute Filename	:	note pad.exe
Option	:	%f
Executing Directory	:	%x
Designate Additional Option when Executing:		A check mark is not set.
Use Output Window	:	A check mark is not set.

-When the dir command is registered

Title	:	Dir
Execute Filename	:	command.com
Option	:	/c dir
Executing Directory	:	
Designate Additional Option when Executing:		A check mark is not set.
Use Output Window	:	A check mark is set.

4.7.5 Keyboard

"Keyboard..." enables definition of shortcut keys.

■ Keyboard setup procedure

1. Select a type.
   Functions are displayed in [Function List].
2. Select the function to be set from [Function List].
   The explanation of the selected function is displayed in the explanation field (lower part) of the keyboard setup dialog box. When an assigned function is selected, the currently assigned keys are displayed in [Assign key].
3. Set a focus in [New Assign], then specify the key to be assigned to the selected function from the keyboard (press the key).
4. Click the [Set] button.

■ Procedure for deleting an assigned key

1. Select the function corresponding to the key to be deleted (see (1) and (2) in the keyboard setup procedure above).
   The currently assigned keys are displayed in [Assign key].
2. Select the key to be deleted from the key list displayed in [Assign key].
3. Click the [Delete] button.
   The dialog box for checking to be deleted the key opens.
4. Click the [OK] button.

■ Procedure for changing an assigned key

1. Delete an assigned key (see the procedure for deleting an assigned key above).
2. Set a focus in [New Assign], then specify the key to be assigned to the function from the keyboard (press the key).
3. Click the [Set] button.

■ Displaying the current setup state list

Click the [Definition List] button to display the key definition list.

■ Restoring all the set keys to the initial state

Click the [Reset] button.

Note:

-Several keys can be assigned to one function. In this case, the assigned keys have the same function.

-Once the [Set] or [Reset] button is clicked, the set or reset key cannot be canceled. If the [Set] or [Reset] button is clicked by mistake, set the key again.

-As for the keys that can be set, see Table 4.7-2.

Figure 4.7-40 Keyboard Setup Dialog Box

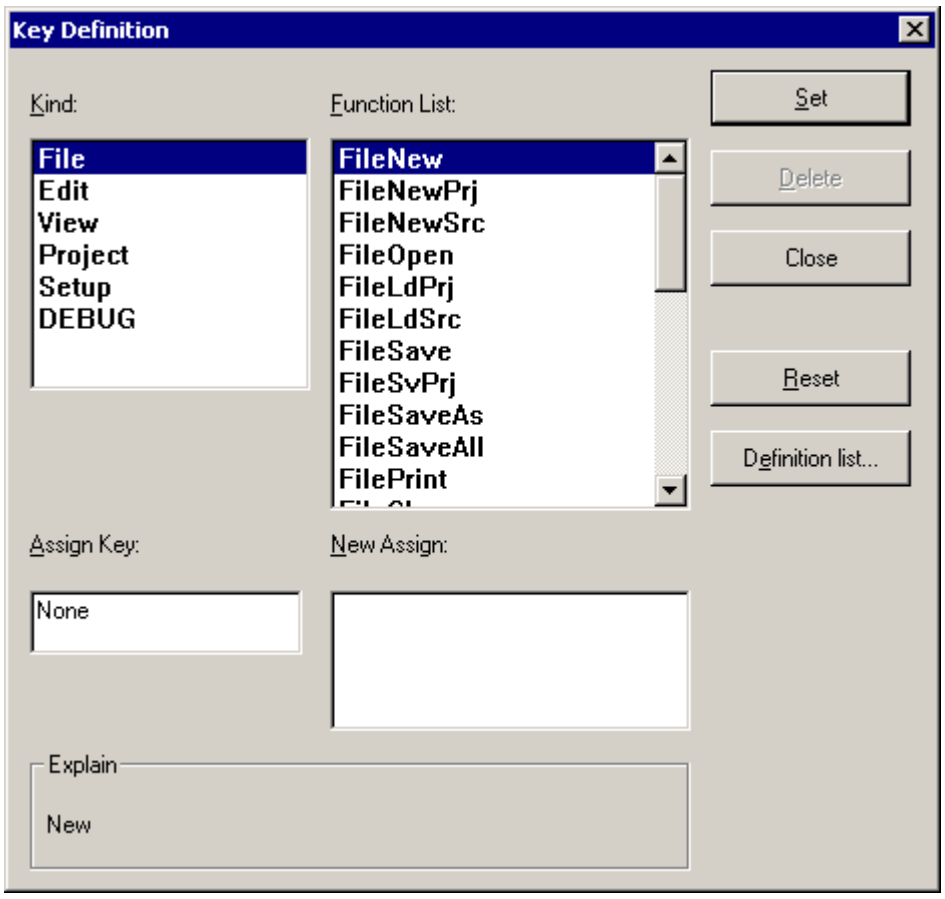


Table 4.7-2 Keys That can be Set

Key	Explanation
CTRL + A to Z	Press any of the A to Z keys while holding down the CTRL key.
SHIFT + CTRL + A	Press any of the A to Z keys while holding down the SIFT and CTRL keys.
SHIFT + F1	Press the F1 key while holding down the SHIFT key.

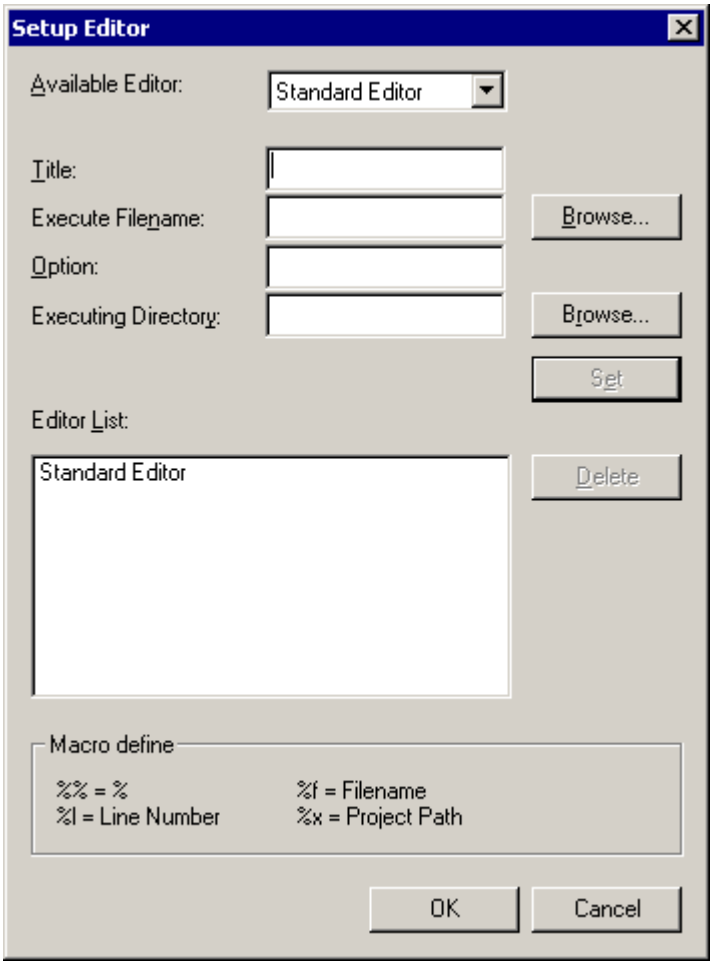
4.7.6 Editor

"Editor..." enables any editor to be registered and used as the standard editor.

■ Registering an editor

Register the editor to be used instead of the standard editor built into SOFTUNE WORKBENCH in advance. Set the registered editor as the SOFTUNE WORKBENCH editor before editing the file actually. Of the registered editors, the editor set in [Usable Editor] is used to edit the file.

Figure 4.7-41 Editor Setup Dialog Box



■ Editor registration procedure

1. Select [Editor...] from the [Setup] Menu.
   The editor setup dialog box shown in Figure 4.7-41 opens.
2. Set a unique title that differs from the registered names.
3. Specify the execution file name of the editor to be registered.
   Clicking the [Browse] button to the right of this field opens the file selection dialog box. The execution file name of the editor can be selected from this dialog box.

4. Set an option.
   Macro description can be used in this field. For macro description, refer to Section "1.9 Storing External Editors_ in SOFTUNE WORKBENCH User's Manual".
5. Write a run-time directory.
   This description may be omitted if control need not to be moved to any specific run-time directory.
   Clicking the [Browse] button to the right of this field opens the file selection dialog box. A run-time directory can be selected from this dialog box.
6. Click the [Set] button.

■ Editor deletion procedure

1. Select [Editor...] from the [Setup] Menu.
   The editor setup dialog box shown in Figure 4.7-41 opens.
2. Select the title of the editor to be deleted from the editor list.
3. Click the [Delete] button.

■ Editor change procedure

1. Select [Editor...] from the [Setup] Menu.
   The editor setup dialog box shown in Figure 4.7-41 opens.
2. Select the title of the editor to be changed from the editor list.
   The values set in [Title], [Execute Filename], [Option], and [Executing Directory] are displayed.
3. Change [Execute Filename], [Option], and [Executing Directory]. When [Title] is changed, the set editor is registered as another editor.
4. Click the [Set] button.
   The dialog box asking you whether to change the editor opens.
5. Click the [Yes] button.

■ Setting the editor to be used

1. Register the external editor to be used according to the editor registration procedure.
2. Click the [▼] button to the right of the [Available Editor] field. The drop-down list showing registered editor titles is displayed.
3. Select the editor title to be used from the drop-down list.

■ Example

Example of Fujitsu Power EDITOR setup

Title	:	Power EDITOR
Execution File Name	:	c:\Powered\powered. exe
Option	:	"%f"-g%l
Run-Time Directory	:	%x

Entering the above and clicking the [Setup] button, registers Fujitsu Power EDITOR in the editor list.

After registering Fujitsu Power EDITOR, select [Power EDITOR] from [Available Editor] and click the [OK] button.

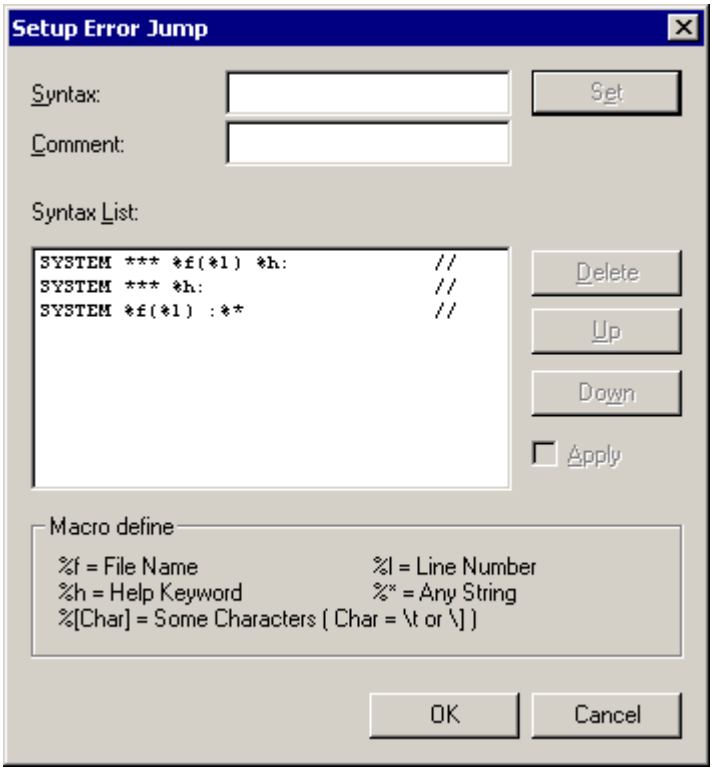
4.7.7 Error

"Error..." registers error message patterns of various tools to enable error jump.

■ Error jump setup procedure

1. Select [Error...] from the [Setup] Menu.
   The error jump setup dialog box shown in Figure 4.7-42 opens.
2. Enter a syntax.
   For details on syntaxes, refer to Section "1.7 Error Jump Function" in SOFTUNE WORKBENCH User's Manual.
3. Enter a comment as required.
   A comment can be added to each syntax.
4. Click the [Set] button.

Figure 4.7-42 Error Jump Setup Dialog Box



■ Syntax deletion procedure

1. Select [Error...] from the [Set] Menu.
   The error jump setup dialog box shown in Figure 4.7-42 opens.
2. Select the syntax to be deleted from the syntax list.
3. Click the [Delete] button.

■ Syntax modification procedure

Modify a set syntax in the following procedure:

1. Select [Error...] from the [Setup] Menu.
   The error jump setup dialog box shown in Figure 4.7-42 opens.
2. Select the syntax to modified from the syntax list.
   The syntax and comment are displayed in the associated fields.
3. Modify the syntax and comment, then click the [Set] button.
   The modified syntax and comment are newly set.
4. Delete an unnecessary syntax (syntax used before modification).

■ Analysis order change and application ON/OFF

-Analysis order change

Error messages are analyzed from the patterns registered in the upper part of the syntax list. To assure correct analysis, the analysis order may have to be changed. The analysis order can be changed in the following procedure:

1.Select [Error...] from the [Setup] Menu.
  The error jump setup dialog box shown in Figure 4.7-42 opens.

2.Select the syntax whose order is to be changed from the syntax list.

3.Click the [UP] or [Down] button to move the cursor to the position where error jump is to be set.

-Application ON/OFF

When the check mark to the left of [Apply] is not set, error messages are not analyzed according to the registered syntax.

When a check mark is set to the left of [Apply], ON is displayed in the syntax list. When a check mark is not set, OFF is displayed in the syntax list.

■ Example of error jump setup

When the error format is [error-message : line-number file-name]

Syntax	:	%* : %1 %f
Comment	:	sample

Note:

The syntax for which SYSTEM is displayed in the syntax list cannot be deleted.

4.7.8 Tool Startup

This section explains how to start a registered external tool.

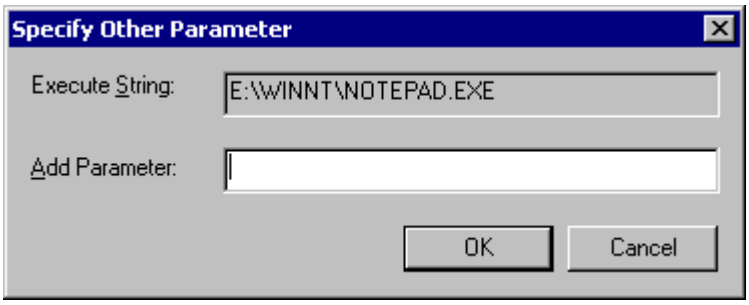
■ Starting an external tool

The tools set by Section "4.7.4 Tool" are registered in the submenu. A tool can be started by selecting it from this submenu.

Setting a check mark to the left of [Designate Additional Option when Executing] in tool setup opens the additional option setup dialog box shown in Figure 4.7-43 before the tool is started. Set an additional option from this dialog box, then click the [OK] button.

The option specified from this dialog box is added after the option specified in tool setup and the tool is started.

Figure 4.7-43 Additional Option Setup Dialog Box



4.8 Window

"Window" controls window display.

■ Control related to window display

- Cascade
- Vertical
- Horizon
- Split
- Arrange Icons
-Refresh Window
-Refresh All Windows
-Close All Windows

■ Window name display

Up to 9 currently open window names are displayed, including icon windows. If ten windows or more are open, the tenth and subwindows are displayed in [Other Windows].

4.8.1 Cascade, Vertical, Horizon

"Cascade", "Vertical", and "Horizon" specify the display formats of subwindows (e.g., Source Window, Register Window, and Assembly Window).

■ Cascade

"Cascade" displays currently open subwindows in the main window.

■ Vertical

"Vertical" arranges currently open subwindows vertically and fully displays them in the main window.

■ Horizon

"Horizon" arranges currently open subwindows horizontally and fully displays then in the main window.

4.8.2 Split

"Split" specifies where a window is vertically split.

■ Split

"Split" specifies where a window is vertically split. The following windows can be vertically split.

-Source Window

-Disassembly Window

-Trace Window

-Memory Window

-Coverage Window

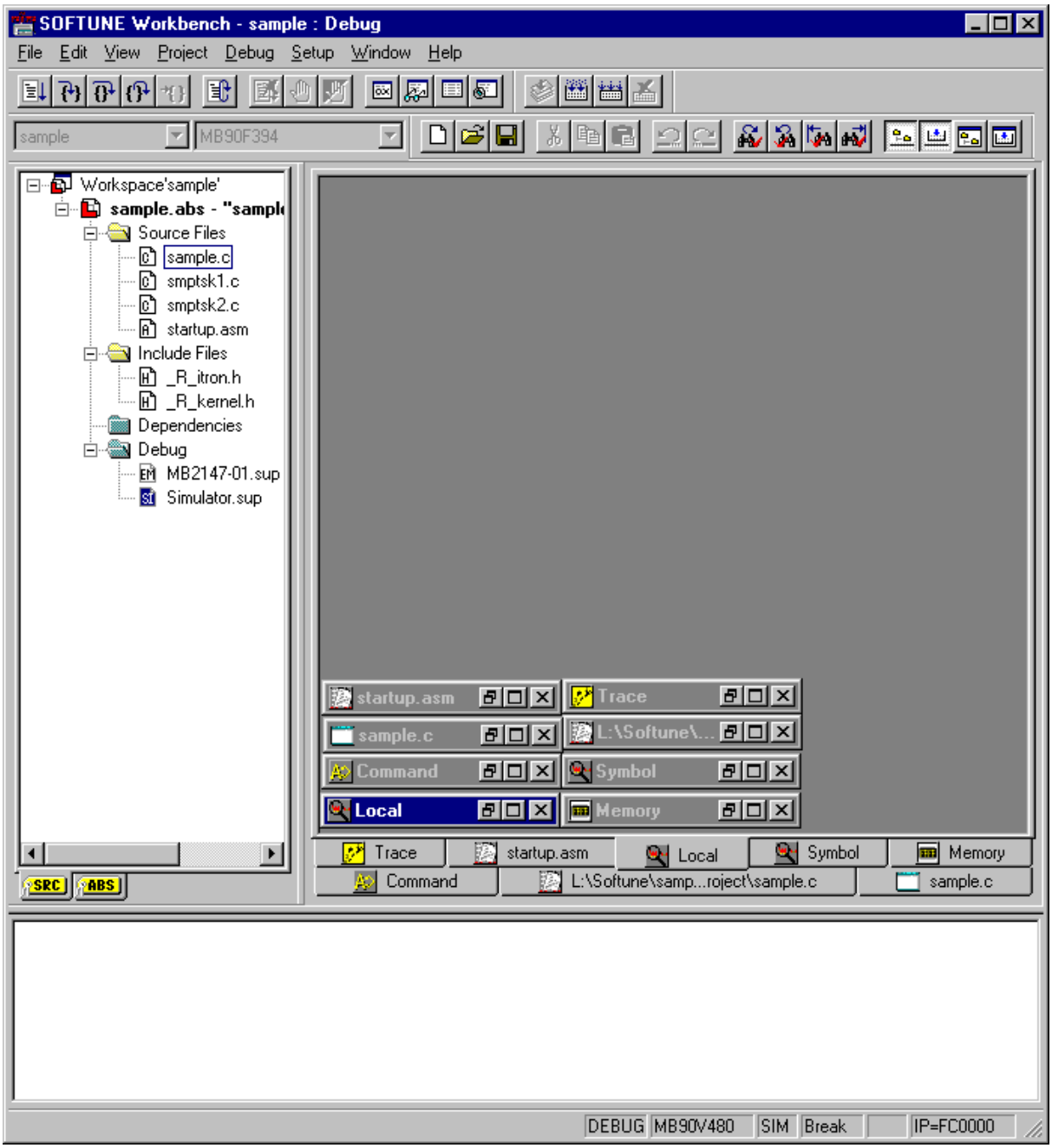
4.8.3 Arrange Icons

"Arrange icons" arranges the locations of the minimized window icons.

■ Icon arrangement

"Arrange icons" arranges all the minimized windows in the SOFTUNE WORKBENCH main window (Figure 4.8-1). However, unminimized windows are not affected.

Figure 4.8-1 Main Window State after Icon Arrangement



4.8.4 Refresh Window

This command updates information on an active window.

■ Refresh window

Information on the current active window is updated.

4.8.5 Refresh All Windows

This command updates information on all the open windows.

■ Refresh all windows

Information on all the open windows except the SRC tab of Project and Output windows is updated.

4.8.6 Close All Windows

"Close all windows" closes all open windows.

■ Close all windows

"Close all windows" closes all currently open windows other than the Project and Output Windows. If the File Edit Window being edited has not saved yet, the dialog box asking you whether to save the window opens.

The file opened by the external editor cannot be closed by this function.

Even if all windows are closed, the SOFTUNE WORKBENCH state remains unchanged. For this reason, register values, etc., are not affected even during debugging.

4.9 Help

"Help" displays online help.

■ Online help

-Help Topics

-Support Information

■ Version information

-Version Information

4.9.1 Help Topics

"Help Topics" retrieves help items according to keywords.

■ Contents

"Contents" hierarchically displays online help contents. It is used to search the contents for the item to be searched.

■ Keyword

SOFTUNE WORKBENCH searches the help file for the directly specified item.

4.9.2 Support Information

"Support Information" opens the attached support information file in the Edit Window.

■ Support information

Support information provides the information not written in the attached manual. Please read through support information once before using SOFTUNE WORKBENCH.

4.9.3 About Fs907s...

"About Fs907s..." displays SOFTUNE WORKBENCH version information.

■ Version information display when SOFTUNE WORKBENCH is not in the debug session

"About Fs907s..." opens the version information dialog box showing the SOFTUNE WORKBENCH logo mark and version information.

■ Version information display when SOFTUNE WORKBENCH is in the debug session

"About Fs907s..." opens the version information dialog box showing the SOFTUNE WORKBENCH logo mark, version information, type of the currently selected debugger, and type of the target MCU being debugged.

Reference:

Be sure to let us know the displayed version information when asking our company about SOFTUNE WORKBENCH.

CHAPTER 5
Addin Module

This chapter explains SOFTUNE WORKBENCH Addin module.

5.1 Customize Bar

5.1 Customize Bar

This chapter describes the Customize Bar of the SOFTUNE Workbench.

5.1.1 What is Customize Bar?
5.1.2 Customize Bar Menu
5.1.3 Registering in the Customize Bar
5.1.3.1 Registering Batch File
5.1.3.2 Registering Workbench Menu
5.1.3.3 Registering External Tool
5.1.4 Warning and Error Messages
5.1.5 Note

5.1.1 What is Customize Bar?

This section describes the customize bar and how to install it.

■ What is Customize Bar?

The customize bar registers batch files, Workbench menus, and external tools used while running the Debugger in a tool bar. This function enables you to call them up easily by clicking and external tool one of the buttons.

You can register up to ten objects. Items that have been set once are restored when the Workspace is opened.

Figure 5.1-1 Customize Bar

By registering batch files or Workbench menus or external tools in the customize bar, the button of the registered number is enabled when the Workspace is opened.

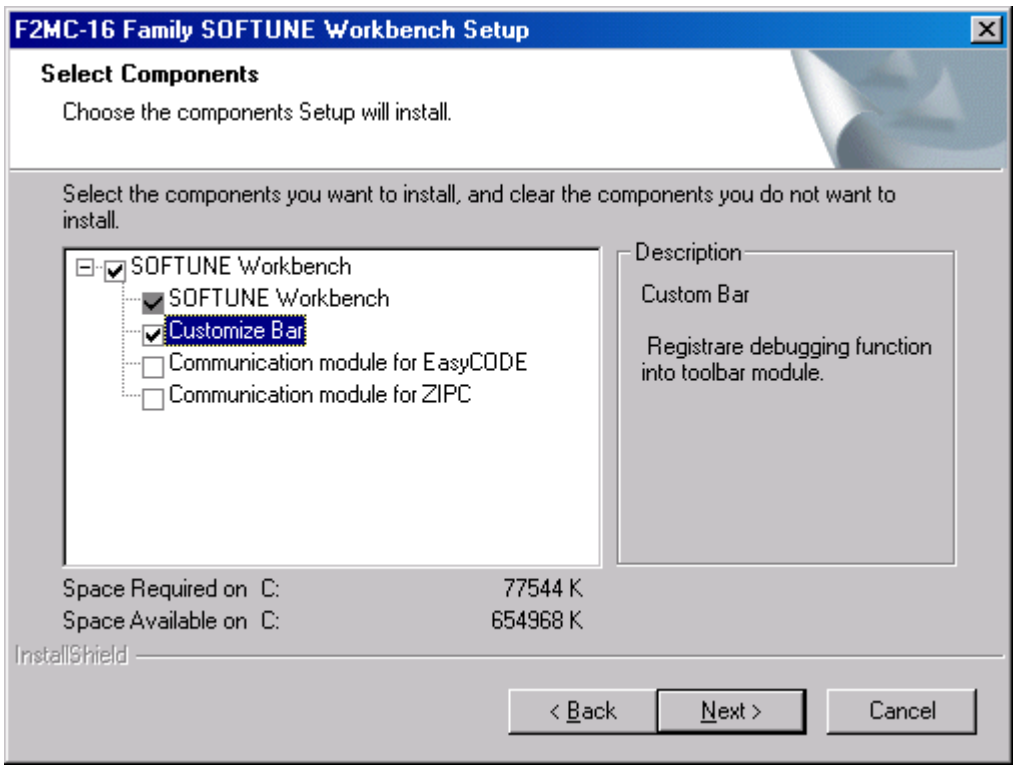
Also, currently registered items are displayed by moving the mouse cursor over any button in the tool bar.

■ Installing Customize Bar

Apply a check mark to the "Customize Bar" in the dialog box (Figure 5.1-2) that is displayed when installing SOFTUNE Workbench to install the customize bar.

Note that you can also install only the customize bar if it was not installed when you installed SOFTUNE Workbench.

Figure 5.1-2 Dialog Box Displayed When Installing


F2MC-16 Family SOFTUNE Workbench Setup
Select Components
Choose the components Setup will install.
Select the components you want to install, and clear the components you do not want to install.
SOFTUNE Workbench
SOFTUNE Workbench
Customize Bar
Communication module for EasyCODE
Communication module for ZIPC
Description
Custom Bar
Registrare debugging function into toolbar module.
Space Required on C: 77544 K
Space Available on C: 654968 K
InstallShield
< Back
Next >
Cancel

When the customize bar is installed, the "Customize Bar" is added to the SOFTUNE Workbench [View] menu (Figure 5.1-3) and a tool bar (Figure 5.1-1) for the customize bar is displayed.

5.1.2 Customize Bar Menu

This section describes the customize bar menu.

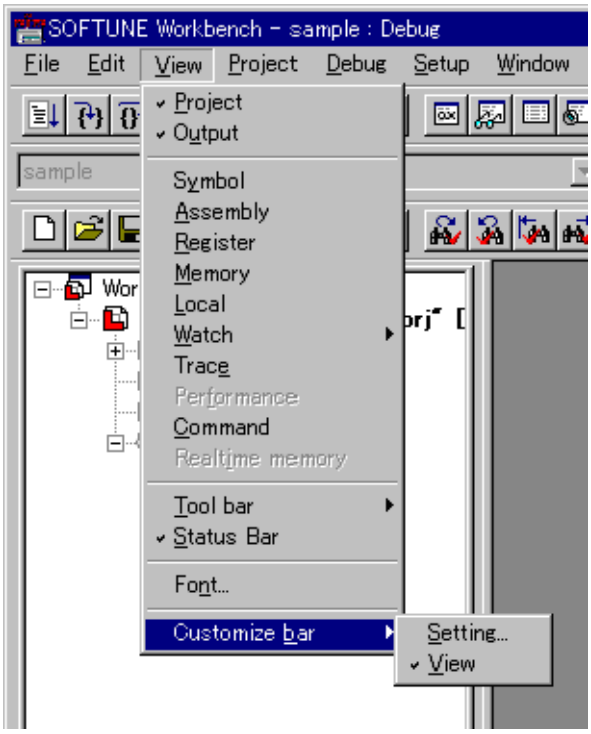
■ Customize Bar Menu

There are two submenus in the [Customize Bar].

Setting: Registers batch files and Workbench menus in the customize bar.
This menu is enabled when opening a workspace.

View: Switches to view/hide the tool bar for the customize bar.
This menu is always enabled when SOFTUNE Workbench is running.

Figure 5.1-3 Customize Bar Menu



5.1.3 Registering in the Customize Bar

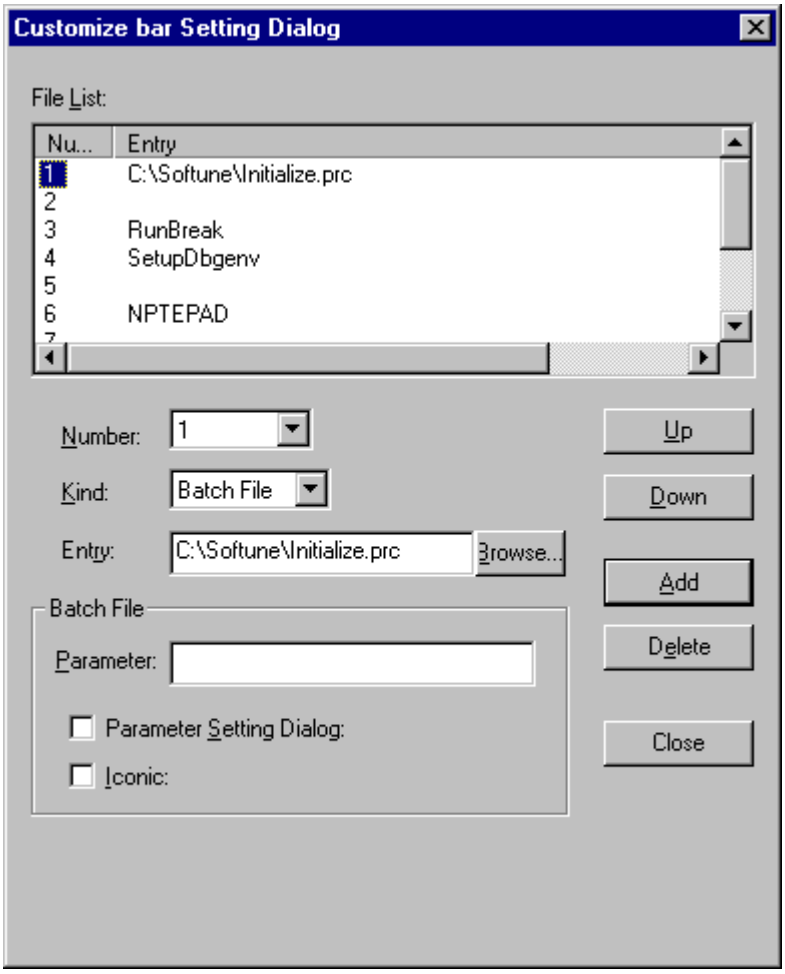
This section describes registering in the customize bar.

■ Registering in Customize Bar

You can registers "batch files" and "Workbench menus" and "external tools" in the customize bar. Register using the "Customize Bar Setting Dialog" (Figure 5.1-4) displayed in [View] - [Customize Bar] - [Setting].

For details on how to register, see Section "5.1.3.1 Registering Batch File", "5.1.3.2 Registering Workbench Menu" and "5.1.3.3 Registering External Tool".

Figure 5.1-4 Customize Bar Setting Dialog



■ Items in Dialog Box

- File List

  The settings registered in the customize bar are displayed.

- Number

  Specify the number to register in the customize bar. Numbers from 1 to 10 can be specified.

- Kind

  Select either "Batch File" or "Menu" to register in the customize bar.

- Entry

  1. When "Batch File" is selected in "Kind"

     Enter the batch file name to be registered in the customize bar. To select from a list, specify with the "Open File" Dialog Box (Figure 5.1-5) displayed when you click the reference button on the right.

  2. When "Menu" is selected in "Kind"

     Enter the Workbench menu to be registered in the customize bar. Specify with the "Menu List" Dialog Box (Figure 5.1-7) displayed when you click the reference button on the right.

  3. When "Tool" is selected in "Kind"

     Enter the external tool to be registered in the customize bar. Specify with the "Select Tool" Dialog Box (Figure 5.1-10) displayed when you click the reference button on the right.

- Parameter (Enabled only when "Batch File" selected in "Kind")

  Input the parameter for executing a batch file.

- Parameter Setting Dialog (Enabled only when "Batch File" selected in "Kind")

  Displays "Parameter Setting Dialog" (Figure 5.1-6) that can set/change parameters with the customize bar when executing a batch file.

- Iconic (Enabled only when "Batch File" selected in "Kind")

  This makes Workbench an icon when executing a batch file with the customize bar.

- Function Explain (Enabled only when "Menu" selected in "Kind")

  Displays a description of the Workbench menu to be registered in the customize bar.

- Up

  Changes the order of the registered contents displayed in the "File List" to one above. Switches that order when there is already one registered to a number one above.

- Down

  Changes the order of the registered contents displayed in the "File List" to one below. Switches that order when there is already one registered to a number one below.

- Add

  Adds batch files, Workbench menu or External tool to the customize bar. If an item has already been registered in the specified number, that number will be rewritten and registered.

- Delete

  Deletes the contents registered in the customize bar. Specify the number to delete in the "File List".

Figure 5.1-5 Dialog Box for Open File

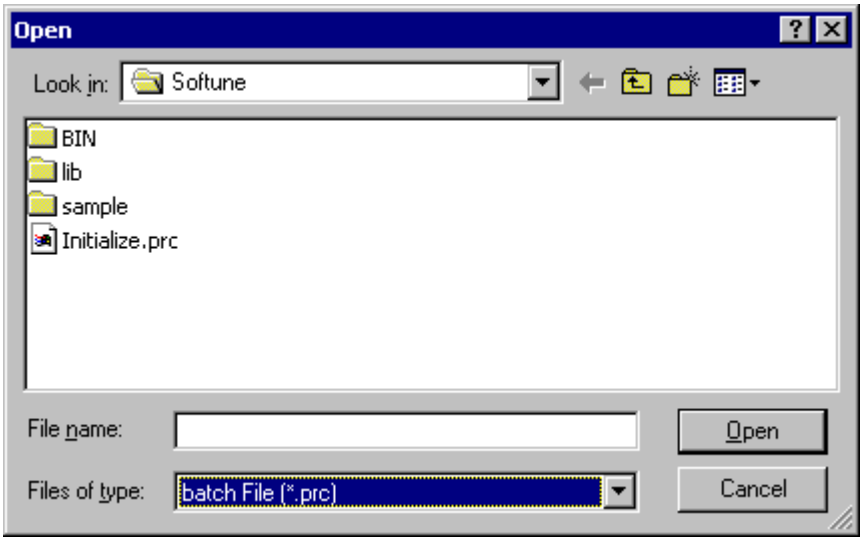


Figure 5.1-6 Parameter Setting Dialog

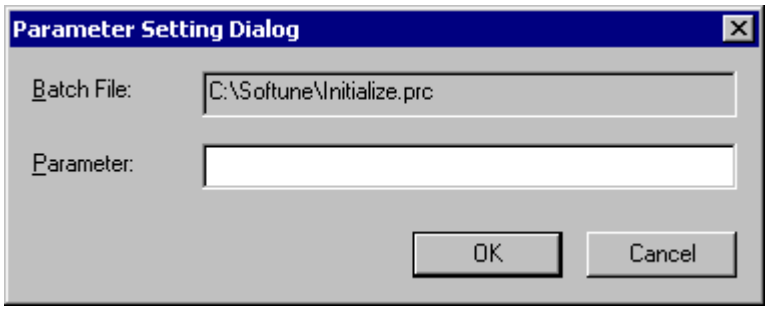


Figure 5.1-7 Menu List Dialog

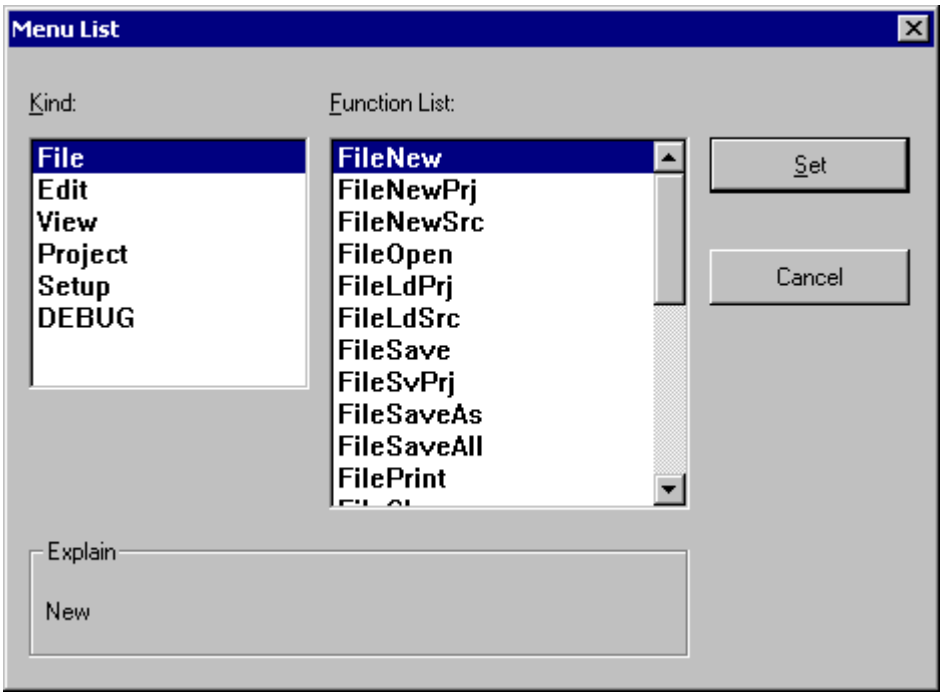


Figure 5.1-8 Select Tool Dialog

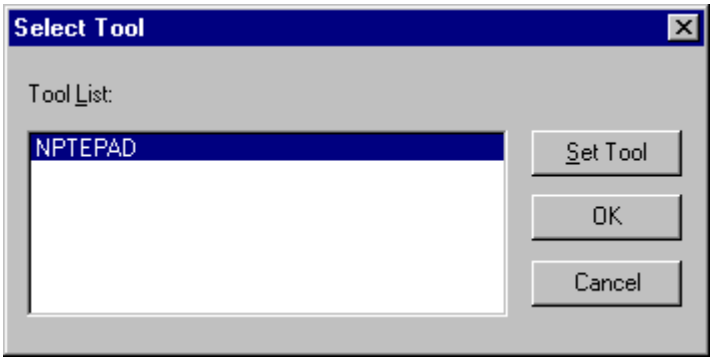
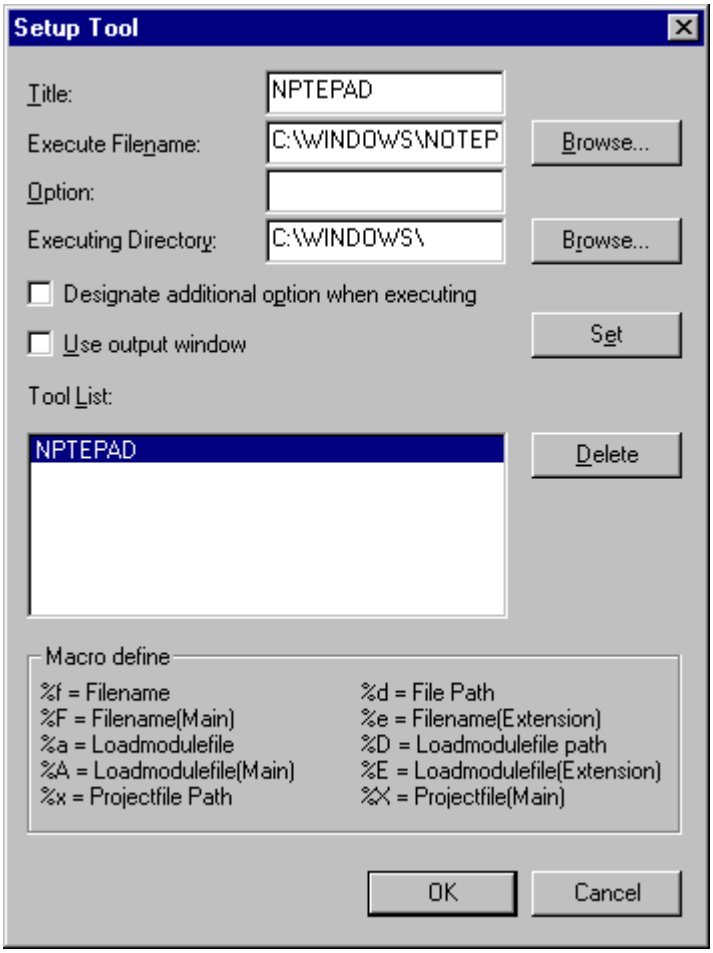


Figure 5.1-9 Setup Tool Dialog



5.1.3.1 Registering Batch File

This section describes registering a batch file in the customize bar.

■ How to register batch file

1. Display dialog box

   Select [View] - [Customize Bar] - [Setting] to display the "Customize Bar Setting" dialog box (Figure 5.1-10).

2. Number

   Select the number to be registered in the customize bar.

3. Kind

   Select "Batch File".

4. Entry

   Enter the batch file name to register in the customize bar. You can specify using the "Open File" dialog box (Figure 5.1-5) displayed when you click the reference button on the right.

5. Parameter

   Input here when specifying a parameter for a specified batch file. Use a comma to separate multiple parameters.

6. Display the "Parameter Setting" dialog box (Figure 5.1-6).

   To set a parameter when executing a batch file with the customize bar, apply a check mark to this. This is convenient when you want to specify/change parameters each time you execute a batch file.

7. Iconic

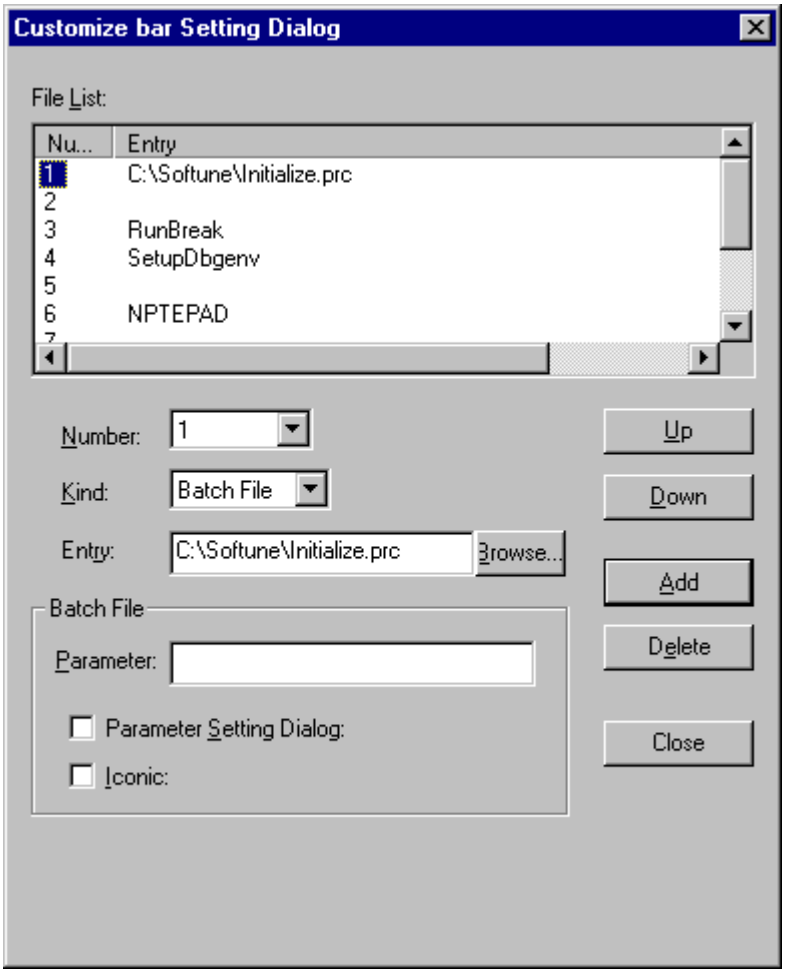
   To make Workbench an icon when executing a batch file with the customize bar, apply a check mark to this.

8. Add

   Check the input contents of 2 to 7. If they are correct, click "Add".

This completes the registration of a batch file. The contents of the registration are displayed in "Settings" and are restored when the Workspace is opened.

Figure 5.1-10 Setting Dialog Box - Batch File



5.1.3.2 Registering Workbench Menu

This section describes how to register the Workbench menu in the customize bar.

■ How to Register Workbench Menu

1. Display the dialog box

   Select [View] - [Customize Bar] - [Setting] to display the "Customize Bar Setting" dialog box (Figure 5.1-11).

2. Number

   Select the number to be registered in the customize bar.

3. Kind

   Select "Menu".

4. Entry

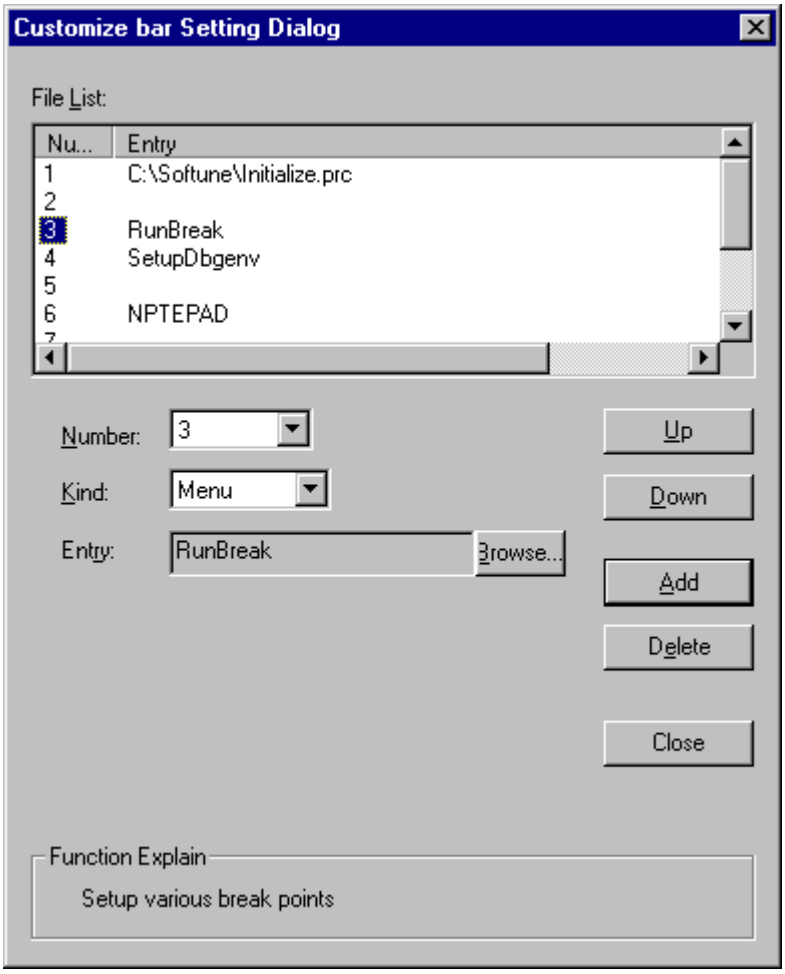
   Enter the Workbench menu to register in the customize bar. Specify with the "Menu List dialog box (Figure 5.1-7)" displayed when you click the reference button on the right.

5. Add

   Check the input contents of 2 to 4. If they are correct, click "Add".

This completes the registration of Workbench menus. The registered contents are displayed in "File List" and are restored when the Workspace is opened.

Figure 5.1-11 Setting Dialog Box -Menu



5.1.3.3 Registering External Tool

This section explains how to register the external tool in the customize bar.

■ How to Register External Tool

1. Display the dialog box

   Select [View] - [Customize Bar] - [Setting] to display the "Customize Bar Setting" dialog box (Figure 5.1-12).

2. Number

   Select the number to be registered in the customize bar.

3. Kind

   Select "Tool".

4. Entry

   Enter the external tool to be registered in the customize bar. Specify with the "Select Tool" Dialog Box (Figure 5.1-10) displayed when you click the reference button on the right.

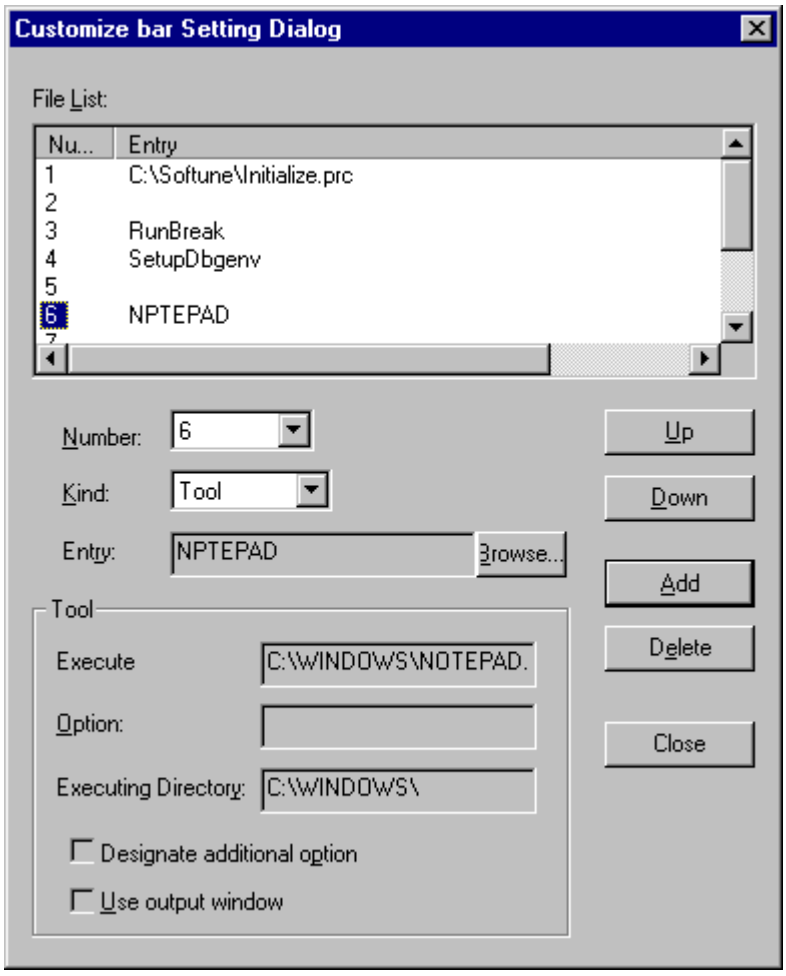
   Select tool dialog shows the contents tools that are set with the "Set Tool" Dialog Box (Figure 5.1-11) displayed when you select [Environment] - [Setup tool] menu. Alternatively, you can display "Set Tool" Dialog Box by the "Set tools" button in the "Select Tool" Dialog Box.

5. Add

   Check the input contents of 2 to 4. If they are correct, click "Add".

This completes the registration of the external tools. The registered contents are displayed in "Settings" and are restored when the Workspace is opened.

Figure 5.1-12 Setting Dialog Box - External Tool



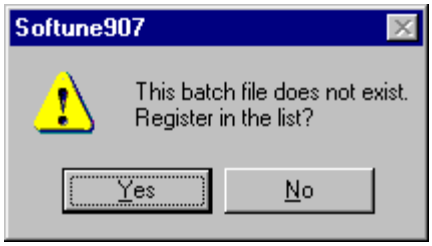
5.1.4 Warning and Error Messages

This section describes the warning and error messages displayed when using the customize bar.

■ Warning Message

1. The following warning message (Figure 5.1-13) is displayed when you click "Add" regardless of whether an input batch file does not exist in the "Entry", when registering a batch file in the customize bar. (See Section "5.1.3.1 Registering Batch File".)

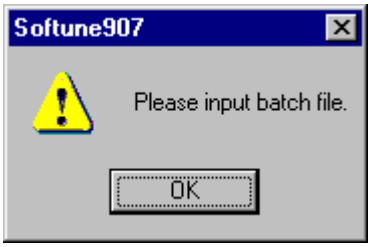
Figure 5.1-13 Warning Message



■ Error Messages

1. The following error message (Figure 5.1-14) is displayed when you click "Add" without entering a batch file in the "Entry", when registering a batch file in the customize bar. (See Section "5.1.3.1 Registering Batch File".)

Figure 5.1-14 Error Message 1



2. The following error message (Figure 5.1-15) is displayed when you click "Add" without entering a machine name in the "Entry", when registering the Workbench menus in the customize bar. (See Section "5.1.3.2 Registering Workbench Menu".)

Figure 5.1-15 Error Message 2

3. The following error message (Figure 5.1-16) is displayed when you click "Add" without entering a tool name in the "Entry", when registering the External tools in the customize bar (See Section "5.1.3.3 Registering External Tool").

Figure 5.1-16 Error Message 3

5.1.5 Note

This section describes the precautions for using the customize bar.

■ Note

1. The customize bar cannot be used when the workspace is not opening. When opening the workspace, the previous settings are restored and the customize bar buttons are enabled.
2. Customized bars for registration are stored for each tool. This allows a common set of restored data to be provided even if projects, workspaces or setup files are switched.
3. When registering batch files in the customize bar, input the relative path or the absolute path from the current directory (= project directory) for the batch files.
4. When registering batch files in the customize bar, always use a comma to separate parameters when specifying multiple parameters. This is the same for the "Parameter Setting" dialog box (Figure 5.1-6).
5. When you registered batch files or Workbench menu that becomes valid when activating the debugger in the customize bar, they won't be executed by pressing the button while the debugger is inactive.

APPENDIX

APPENDIX A List of Register Names
APPENDIX B Downloading Monitor Program
APPENDIX C Setting LAN Interface
APPENDIX D Setting USB Interface
APPENDIX E Creating ROM on Monitor Debugger Target
APPENDIX F Building REALOS Debug Modules into Program
APPENDIX G Display on Emulator
APPENDIX H Guide to Changing Function-call Interface
APPENDIX I External I/F DLL for Simulator

APPENDIX A List of Register Names

Register names are displayed.

■ Registers

The registers that can be operated by SOFTUNE Workbench differ for each MCU type as follows:

Accumulator	:	A
Status register	:	PS
Stack pointers	:	USP, SSP
Bank registers	:	DTB, USB, SSB, ADB
General-purpose registers:		RW0 to RW7
Others	:	RP, ILM, DPR, CCR
Flags	:	C, V, Z, N, T, S, I

The following registers can be also used as special registers.

IP	:	PCB:24-bit PC register
SP	:	24-bit register of the current stack (This register differs from the register in the chip specification)

APPENDIX B Downloading Monitor Program

To use the emulator debugger, the monitor program corresponding to the chip to be used must be written to the emulator.
This processing is called "monitor program download".
For MB2147-01 and MB2147-05 emulator, data in the emulator can be checked at the beginning of debugging to automatically download the appropriate emulator monitoring program and version data into the emulator.

■ Downloading Monitor Program

- When the emulator is the MB2141 series

The Downloading Monitor Program procedure is described below:

1.Connect the emulator to a personal computer (PC) with an RS-232C or LAN interface.
(When connecting the emulator to the PC, see APPENDIX C Setting LAN Interface.)

2.Press the reset switch, then turn on the emulator.
Check that the READY LED of the emulator body turn on.

3.Execute the [Monitor Loader] Menu from [SOFTUNE V3] of [FFMC-16 Family SOFTUNE WORKBENCH Tools] of the Windows Start Menu.
The monitor loader program is started.

4.Select the monitor program to be loaded.
Select the monitor program corresponding to the chip to be used.

5.Specify a communication type.
To use the RS-232C interface, specify a communication port and a baud rate.
To use the LAN interface, specify the host name of the emulator.

6.Click [Start Load].
The selected monitor program is downloaded to the emulator.

7.Select [Exit] from the [File] menu to exit the monitor program.

Table B-1 Monitor Program

Chip type	Corresponding chip	Monitor program
F^2MC-16/16H	MB907XX	EML907A.HEX
F^2MC-16F	MB902XX	EML902.HEX(*1) EML902N.HEX(*2)
F^2MC-16L	MB906XX	EML906.HEX(*1) EML906N.HEX(*2)
F^2MC-16LX	MB905XX	EML905.HEX(*1) EML905N.HEX(*2)

*1: For MB2145-506 emulation pod
*2: For MB2145-507 emulation pod

- When the emulator is for the MB2147-01, MB2147-05 series

The Downloading Monitor Program procedure is described below:

Setting the Monitoring Program Automatic Loading is described (3-a)-(5-a), setting the Downloading Monitor Program procedure is described (3-b)-(7-b).

(1-a) Connect the emulator and personal computer using RS-232C LAN interface, or USB.

(When connecting the emulator to the PC, see APPENDIX C Setting LAN Interface.)

(2-a) Press the reset switch, then turn on the emulator.

Check that the READY LED of the emulator body turn on.

(3-a) Execute the [FFMC16 Family SOFTUNE WORKBENCH] menu from [SOFTUNE V3] of the Windows Start Menu.

The SOFUTNE WORKBENCH will start.

(4-a) Create Workspace and Project, start-up the setup wizard.

Select MB2147-01 or MB0147-05 by ICE type, click "Next" button. Check displayed Monitoring Program Automatic Loading check box.

(5-a) Execute the [debug] menu in the [Start debug]. The emulator debugger will startup.

Monitoring program automatically load in "\Installation Directory \LIB\907\"

This will automatically load the monitor program to the emulator at the beginning of debugging.

When download quit, message dialog box ("Load quit correctly") is displayed. If the "OK" button is clicked, then start debug.

(3-b) Execute the [Monitor Loader] menu from the [SOFTUNE V3]-[FFMC16 Family SOFTUNE WORKBENCH Tools] of the Windows Start Menu.

The monitor loader program will be started.

(4-b) Select the monitor program to be loaded.

Select the monitor program corresponding to the chip to be used.

(5-b) Specify a communication type.

To use the RS-232C interface, specify a communication port and a baud rate.

To use the LAN interface, specify the host name of the emulator.

(6-b) Click on [Start Load].

This will load the binary file to the emulator.

(7-b) Select [Exit] from the [File] menu to quit the monitor program.

Note:

Monitor program increase in the MB2147-01 and MB2147-05 emulator. For further details, see release note in CD root directory.

APPENDIX C Setting LAN Interface

To enable LAN communication, the LAN interface must be set at the PC and emulator sides. Consult the LAN administrator when setting the IP address and a port address, etc.

■ Setting LAN interface at PC side

1. Install the TCP/IP protocol in network setting dialog on WindowsXP or Windows2000.
   Install the TCP/IP protocol.
   Click [Control Panel]-[Network and Dial-up Connections]-[Local Area Connection]-[Property] to set a in WindowsXP and Windows2000.
2. Add the IP address, assigned to the emulator, to the HOSTS file.
   Add the following items:
   IP address Host name
   For WindowsXP and Windows2000, the IP address and host name are in SYSTEM32 \DRIVERS\ ETC. Users with administration authority must set the address and name.
3. Register the emulator port address and service name in the SERVICES file. At the default, 5001 is the support address, and fjicesv is the service name. Register the following items:
   fjicesv 5001/tcp
   For WindowsXP and Windows2000, the port address and service name exist in SYSTEM32 \DRIVERS\ ETC. Users, who possess an administrator authority, must set these address and name.

■ Setting LAN interface at Emulator side

In case of MB2141 emulator, this procedure is following:

1. Connect the emulator to the PC with the RS-232C interface.
2. Turn on the emulator.
3. Execute the [LAN Address] Menu from [SOFTUNE V3]- [FFMC-16 Family SOFTUNE WORKBENCH Tool] of the Start Menu.
   The LAN address setup program is started.
4. Select emulator name (MB2141).
5. Click [Set Communication] to set the RS-232C interface.
6.)Click [Read] to read the current emulator setting status.
7. Set IP Address and Port Address. The IP address and port address set values at the PC side are displayed.
8. Usually, [Universal] is used as MAC Address. However, when using [Local] as MAC Address, consult the LAN address administrator.
9. Select [Exit] from the [Setup] menu to exit the LAN address setup program.
10. Press the emulator reset button (button on rear) to reset the emulator.

In case of MB2147-01 emulator, this procedure is following:

1. Connect the emulator to the PC with the RS-232C or USB interface.
2. Turn on the emulator.

3. Execute the [LAN Address] Menu from [SOFTUNE V3]-[FFMC-16 Family SOFTUNE WORKBENCH Tool] of the Start Menu.
   The LAN address setup program is started.
4. Select emulator name (MB2147-01).
5. Click [Set Communication] to set the RS-232C interface.
6. Click [Read] to read the current emulator setting status.
7. Set IP Address, SubNet Mask and Port Address. The IP address, SubNet mask and port address set at the PC side values are displayed.
8. Usually, [Universal] is used as MAC Address. However, when using [Local] as MAC Address, consult the LAN address administrator.
9. Select [Exit] from the [Setup] menu to exit the LAN address setup program.
10. Press the emulator reset button (button on rear) to reset the emulator.

Note:

MB2147-01 have following interface to set IP address, subnet mask and port address:
- LCD of status display on emulator and three setting switch button

This procedure is following:

1) Push ENTER button on emulator to change parameter-input mode.
2) Select " LAN" menu with up or down button and push ENTER button.
3) Select " IP Address" menu with up or down button and push ENTER button.
   Select the digit to change with up or down button, and push ENTER button.
   Change number with up or down button, and push ENTER button.
   To finish input IP address, select "OK" with up button and push ENTER button after input least significant address.
4) Select " Subnet Mask" menu with up or down button and push ENTER button.
   The procedure to input data is as the same as "IP Address").
5) Select " Port Address" menu with up or down button and push ENTER button.
   The procedure to input data is as the same as "IP Address").
6) Select "Exit" menu with up or down button and push ENTER button.

APPENDIX D Setting USB Interface

Communication via USB requires installation of the USB driver in the personal computer.

■ Installation of USB driver

WindowsXP and Windows2000 connect the emulator to a personal computer via USB.

To install the USB driver, proceed as follows:

1. Connect the emulator to a personal computer with the USB cable.
2. When the power supply of the emulator is turned on, the OS requests installation of the USB driver. Specify the directory (Drivers) on this product CD-ROM.

Note:

The following emulator have USB interface.
MB2147-01
MB2147-05

APPENDIX E Creating ROM on Monitor Debugger Target

This appendix explains creation of the monitor debugger target ROM.

- **Target system configuration**
- **Target system creation procedure**
- **Explanation of sample program**

■ Creating ROM on Monitor Debugger Target

To use the monitor debugger, besides the debugger body (mod935.rel, mod911.rel, and mod907.rel), the following program must be added to create a target system:

- Target system initialization routine
- I/O driver used to communicate with host system
- Vector table

Note:

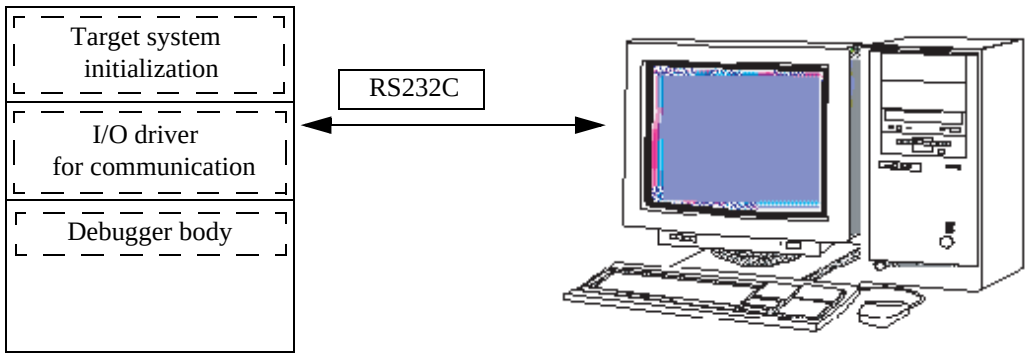
The function-call interface for the REALOS debug module (r_d_dbg.obj) attached to Workbench is created using "stack argument passing".

The target system using "register argument passing" cannot be incorporated.

For details of the function-call interface, see "Appendix H Guide to Changing Function-call Interface".

■ Target system configuration

The monitor debugger configuration is shown in the figure below.



Target system

■ Communication with host system

The target system uses the RS-232C interface to communicate with the host system. For this reason, the target system must be provided with communication hardware.

■ Program suspension (ABORT switch)

The target system should be designed so that a user NMI is issued by pressing the ABORT switch on the user hardware. This design enables the running program to be aborted externally. Prepare this hardware.

■ Target system creation procedure

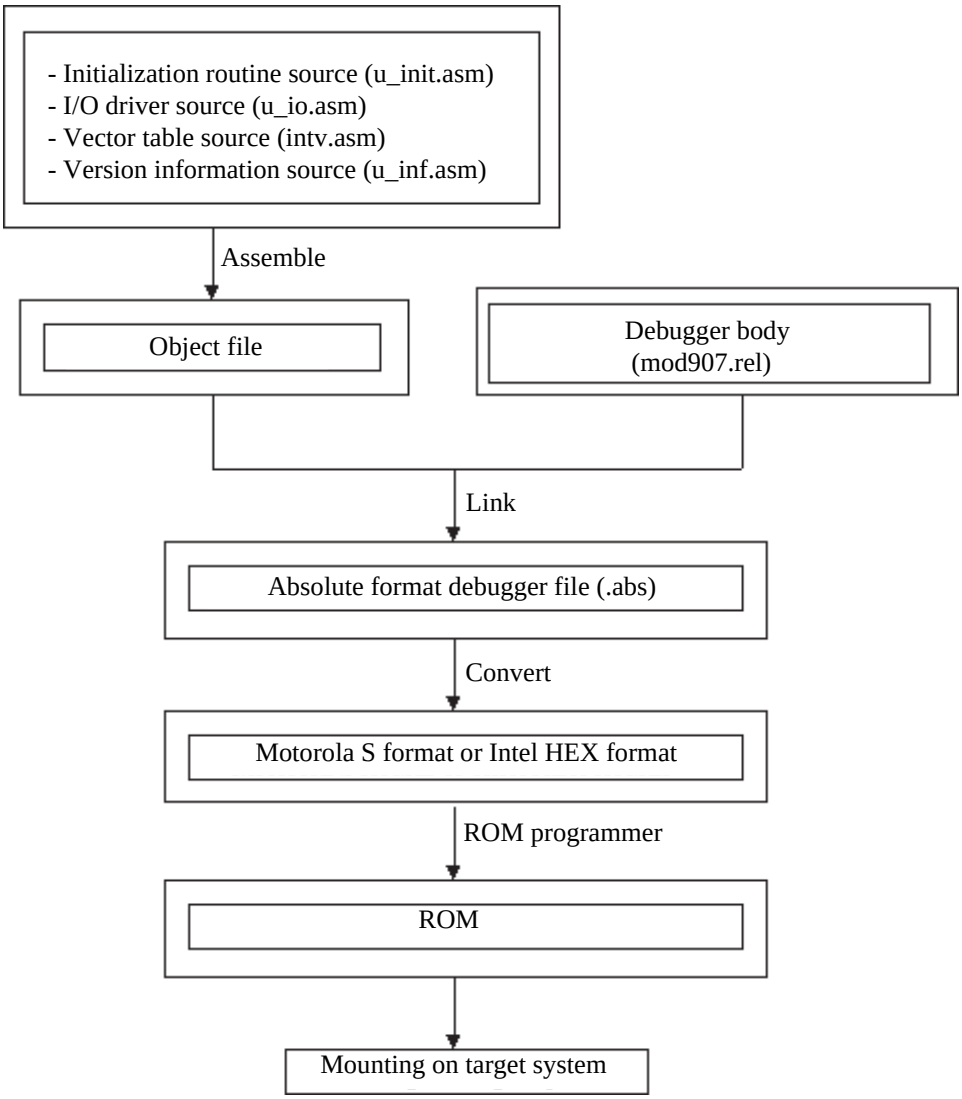
The target system creation procedure is given below. Figure E-1 shows the flow for creating the target system procedure.

1. Create and assemble the initialization routine, I/O drivers, and vector table according to the specifications of the hardware to be used.
2. Link the object file created in (1) to the debugger body (mod907.rel).
3. Convert the absolute format debugger file created in (1) in the Motorola S format or Intel HEX format.
4. Transfer the Motorola S or Intel HEX file created in (3) to the ROM writer. When it receives the file, the ROM writer writes it to ROM.

The monitor debug mod907.rel is in the following directory:

"\Installation Directory \LIB\907\"

Figure E-1 Flow for Target System Creation Procedure



■ Explanation of Sample Program

The SOFTUNE WORKBENCH provides the following files as samples for the initialization routine, I/O driver, and vector table:

- train.inc Definition of I/O addresses of CPU
- sprt670.inc Definition of I/O addresses of CPU
- u_init.asm Initialization (Communication, timer), Setting of stack area.
- u_io.asm Driver (Communication, timer)
- intv.asm Interrupt vector table
- link.opt Input file to linker
- flag.inc Setting of resource select flag, etc.
- mod907.rel monitor unit

■ Sample files

Refer the following files provided as samples the user builds into the target system.

The files to be built into the target system should be rewritten as required.

Some symbol names are fixed because they are externally referenced from the debugger body. Such symbol names are suffixed by an asterisk (*). Use these names as they are.

-sprt670.inc

File for defining labels used by u_io.asm. This file defines F^2MC-16 I/O addresses, etc.

-u_init.asm

- u_init (*)

Initializes communication and timer, etc., used by debugger. When using abort and timer processing, set the 1-byte _abtflg(*) and _timflg(*) flags to 1. When not using them, set the flags to 0.

Set internal ROM or external ROM.

-u_io.asm

- inithrd (*)

Note:

Initializes target system and sets CPU (e.g., memory interface setup)

Do not set them in the user program to be debugged. Depending on the user program to be reset, the debugger may not operate normally.

- initbd

Initializes the evaluation board.

- initrs

Initializes communication ports. Polling control applies to reception and transmission.

- __clr_abrt (*)

When an Abort switch is available, write the processing here when factor clear by software is required.

- _inittime

Initializes measurement timer valid when execution time measurement function ("SHOW TIMER" command) used. If this function is not used, write "RET" only.

- __clrtim (*)

  Clears measurement timer and starts measurement. If the timer function is not used, write "RET" only.

- __readtim (*)

  Reads measurement timer. When not using the timer function, write "RET" only.

- _getchr (*)

  Passes 1-byte receive data to debugger.

- _putchr (*)

  Transmits 1-byte data passed from debugger.

- intv.asm

  Interrupt vector table used by debugger.

- u_inf.asm

  - _user_version_information (*)

    Defines the user individual character string to display in the SOFTUNE Workbench version information.

    V30L25 or more can be used. Up to 31 characters can be defined. Input "\0" (null) at the end of the character string. If display is unnecessary, input only "\0" (null).

- link.opt

  Sample file to be input to linker. This file is used the sample program object is linked to the debugger body (mod907.rel).

Note:

Always link the "STACK" section area and "DATA" section area in this order.
The debugger clears these areas to zero.

Note:

There are cases (communication between host system and target) where the monitor debugger cannot be controlled. The possible causes are:
- Execution is in progress (the debugging state is Execute).
- The target program is running away.

In such cases, restart the debugger as follows:
1. Select [Debug (D)]-[Abort (A)] (or click the Stop button).
2. Click [Abort (A)] in the Abort dialog.
   * In this case, the MCU cannot be reset.
   Ignore the error message even if it appears several times.
3. When the warning "Abort command error" appears, click [OK].
4. Stop the debugger and reset the target system. Restart the debugger.

APPENDIX F Building REALOS Debug Modules into Program

This appendix explains how to build REALOS debug modules into a program to execute REALOS debugging.

■ Building REALOS Debug Modules

The SOFTUNE debugger supports the following REALOS task debug functions.

-Object monitoring

-System call issuing

-Task dispatch breaking

-System call breaking

The object monitoring and system call issuing functions can be used by all programs that the SOFTUNE REALOS debug module is built into.

To use the task dispatch break and system call break functions, debug module 1 (r_d_dbg.obj) must be built into the program.

■ How to build REALOS Debug Modules

When SOFTUNE WORKBENCH uses REALOS debug functions, the REALOS debug module must be built into.

- When REALOS Configurator is used.

Select REALOS debug module on REALOS Configurator and build.

See the SOFTUNE REALOS configurator manual for more detail.

- When REALOS Configurator is not used.

Build R97_DBG.OBJ into the kernel.

-Section.

The code section and data section name of the REALOS debug module are "R97-CODE" and "R97-DATA", respectively.

The F^2MC-16 series has the same section name as that of REALOS, so section arrangements are made together with REALOS.

Notes:

1. Build together with the debug information.
2. The function-call interface for the monitor debugger (mod907.rel) attached to Workbench is created using "stack argument passing".
   The target system using "register argument passing" cannot be configured.
   For details of the function-call interface, see "Appendix H Guide to Changing Function-call Interface".

APPENDIX G Display on Emulator

MB2147-01 have the LCD to display emulator state.

■ LCD display at normal state

The following strings display at normal state:

- Before Debugging

NO CONNECT *

[Enter]: Menu

"NO CONNET" is shown by no connection with host.

At the '*' location, the spinning animation is displayed for waiting for command.

In the second line beginning "[ENTER]", the working to push Enter button is displayed.

- After Debugging

STATUS Vol: *

[Enter]: Menu

In the first line beginning "STATUS", the target status is displayed at "??" location. The status is shown by "EXEC" or "BREAK".

At the '*' location, the spinning animation is displayed for waiting for command.

In the second line beginning "[ENTER]", the working to push Enter button is displayed.

■ Display error at debugging

The following error is displayed, please terminate debugging and turn on the emulator again.

In case of RS-232C communication error, please check the breaking of wire:

- RS-232C Communication Error

  RS232C error

  Check Parameter

- Illegal interrupt error

  System error

  Illegal INT

- REALOS founds abnormal state

  SYSTEM DOWN

  Please RESET

■ Display error at loading monitor program to Emulator

The following error is displayed, please turn on the emulator and load monitor program again:

- Erase Flash Memory Error

  Erase error

  Call to FUJITSU

- Write Flash Memory Error

  WRITE error

  Call to FUJITSU

- Error Occurs at Check RAM at start-up

  RAM error

  Call to FUJITSU

APPENDIX H Guide to Changing Function-call Interface

If a function-call interface is changed, how to use selected registers and how to pass arguments are also changed. Thus, the program will not operate properly if the existing function-call interface (called "stack argument passing") and a new function-call interface (called "register argument passing") are mixed.
Here are the methods and precautions for changing the program to a new function-call interface.

■ New Function-call Interface

The F^{2}MC-16 FAMILY SOFTUNE supports the new function-call interface ("register argument passing") to pass the arguments of the function by registers (RW0 and RW1) as well as the existing function-call interface ("stack argument passing") to pass the arguments by the stack.

Using the function-call interface for "register argument passing", selected arguments are passed by registers, reducing code size and stack usage to improve the program execution speed.

However, the how to pass arguments and to use registers differ between the "register argument passing" and the "stack argument passing." If objects with different function-call interfaces are mixed, the program will not operate properly. Workbench, Assembler, Linker, and Librarian are designed to output warning messages or error messages when detecting a contradiction concerning the function-call interface.

For details about the new function-call interface, refer to the following manuals:

- F^{2}MC-16 FAMILY SOFTUNE C COMPILER MANUAL

  FUNCTION CALL INTERFACE (register argument passing)

  Stack Frame (register argument passing)

  Argument (register argument passing)

  Argument Extension Format (register argument passing)

  Calling Procedure (register argument passing)

  Register (register argument passing)

  Return Value (register argument passing)

- F^{2}MC-16 FAMILY SOFTUNE ASSEMBLER MANUAL

  -rp, -Xrp

  .REG_PASS instruction

- F^{2}MC-16 FAMILY SOFTUNE LINKAGE KIT MANUAL

  Mixing of objects with different function-call interfaces in Linker

  Option (-rp) to specify load module for "register argument passing"

  Option (-Xrp) to specify load module for "stack argument passing"

  Mixing of objects with different function-call interfaces in Librarian

  Option (-rp) to specify output of library for "register argument passing"

  Option (-Xrp) to specify output of library for "stack argument passing"

● F^{2}MC-16 FAMILY SOFTUNE WORKBENCH OPERATION MANUAL

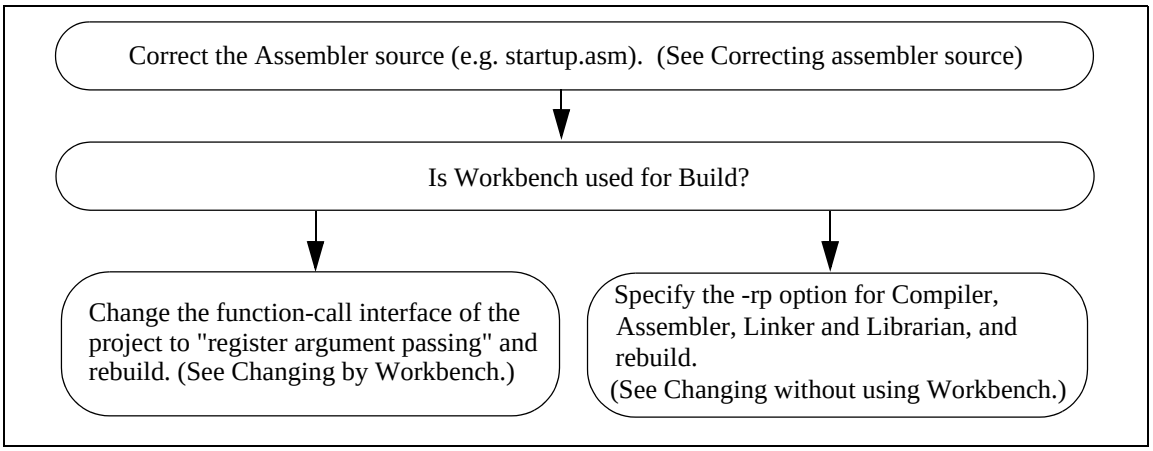
[Storing of Project]

[Setup project] - [General]

■ Changing of Function-call Interface

To change the function-call interface from "stack argument passing" to "register argument passing," use the following procedure:

Figure H-1 Changing of Function-call Interface


Correct the Assembler source (e.g. startup.asm). (See Correcting assembler source)
Is Workbench used for Build?
Change the function-call interface of the project to "register argument passing" and rebuild. (See Changing by Workbench.)
Specify the -rp option for Compiler, Assembler, Linker and Librarian, and rebuild.
(See Changing without using Workbench.)

■ Details of Changing of Function-call Interface

● Correcting assembler source

To change the function-call interface from "stack argument passing" to "register argument passing," use the following procedure to correct the assembler source:

1. Description of .REG_PASS pseudo-instructions

   When the -rp option is specified, Assembler outputs the object for "register argument passing." However, only with the -rp option, Assembler cannot distinguish between the input assembler source for "register argument passing" and the input assembler source for "stack argument passing."

   The .REG_PASS pseudo-instructions indicates the assembler source for "register argument passing."

   When the -rp option is specified, if Assembler assembles the assembler source where the .REG_PASS pseudo-instructions is not described, Assembler will output an error.

   When changing the assembler source for "stack argument passing" to the assembler source for "register argument passing," describe the .REG_PASS pseudo-instructions.

   The .REG_PASS pseudo-instructions can be described anywhere preceding the .END pseudo-instructions.

   [Example of description of .REG_PASS pseudo-instructions]

```
        .PROGRAM    _func
        .TITLE      _func
    ; register argument passing
```

```
.REG_PASS
    :
```

2. Changing the how to pass arguments

   If a function with arguments is defined in the assembler source and is called, change the how to pass arguments to "register argument passing."

   The same correction is also required when defining and calling the function with arguments by the assembler description function (asm statement) of C Compiler.

   When changing the how to pass arguments to "register argument passing," follow the tables in F^{2}MC-16 FAMILY SOFTUNE C COMPILER Manual "Arguments (register argument passing)."

   [Correction example of definitions of functions with arguments]

(stack argument passing)

```
_func:
    LINK    #0
    MOVW    A,    @RW3+4
    ADDW    A,    @RW3+6
    ADDW    A,    @RW3+8
    MOVW    _var, A
            :
```

(register argument passing)

```
_func:
    LINK    #0
    MOVW    RW4,  @RW3+4
    MOVW    A,    RW0
    ADDW    A,    RW1
    ADDW    A,    RW4
    MOVW    _var, A
            :
```

   [Correction example of calling functions with arguments]

(stack argument passing)

```
            :
    MOVW    A,    #3
    PUSHW   A,
    MOVW    A,    #2
    PUSHW   A,
    MOVW    A,    #1
    PUSHW   A,
    CALL    _func
    ADDSP   #6
            :
```

(register argument passing)

```
            :
    MOVW    A,    #3
    PUSHW   A,
    MOVW    A,    #2
    MOVW    RW1,  A
    MOVW    A,    #1
    MOVW    RW0,  A
    CALL    _func
    POPW    AH
            :
```

(Supplementary explanation)

Using the assembler source both for "stack argument passing" and for assembler source for "register argument passing"

Using the predefined macro "__REG_PASS__" create an assembler source both for "stack argument passing" and for "register argument passing". When the predefined macro "__REG_PASS__" is assembled with the -rp option, set 1. When the predefined macro "__REG_PASS__" is assembled without the -rp option, set 0.

[Example of description of predefined macro "__REG_PASS__"]

```
        .PROGRAM    _func
        .TITLE      _func
#if __REG_PASS__
; register argument passing
        .REG_PASS
#endif
                :
```

Note:

With the function-call interface for "register argument passing", return values from the functions of registers RW0 and RW1 are not guaranteed. Do not use these registers across function calls.

● Changing by Workbench

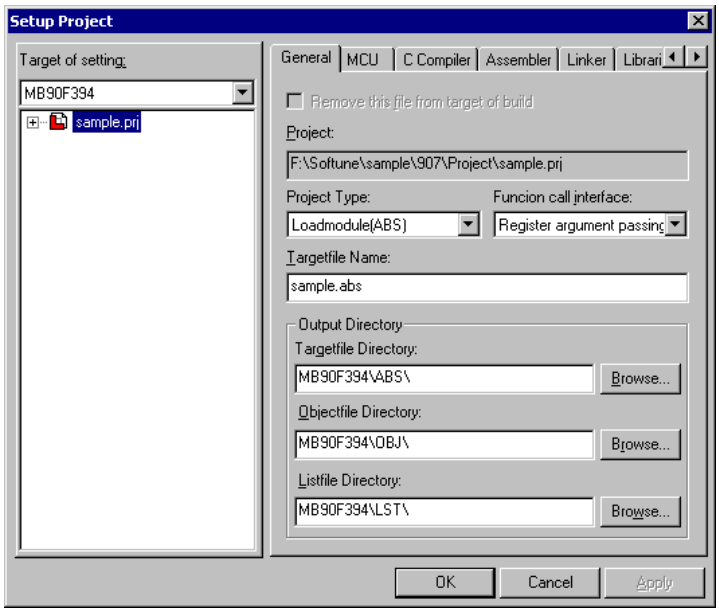
Change the function-call interface of the project to "stack argument passing."

When the function-call interface for "register argument passing" is selected, the -rp option is specified automatically when Compiler, Assembler, Linker, and Librarian start up.

1. Changing the setting of the current project

   Change the function-call interface of the project to "register argument passing" in the [Setup Project] dialog of the project.

Figure H-2 Setup Dialog of Project

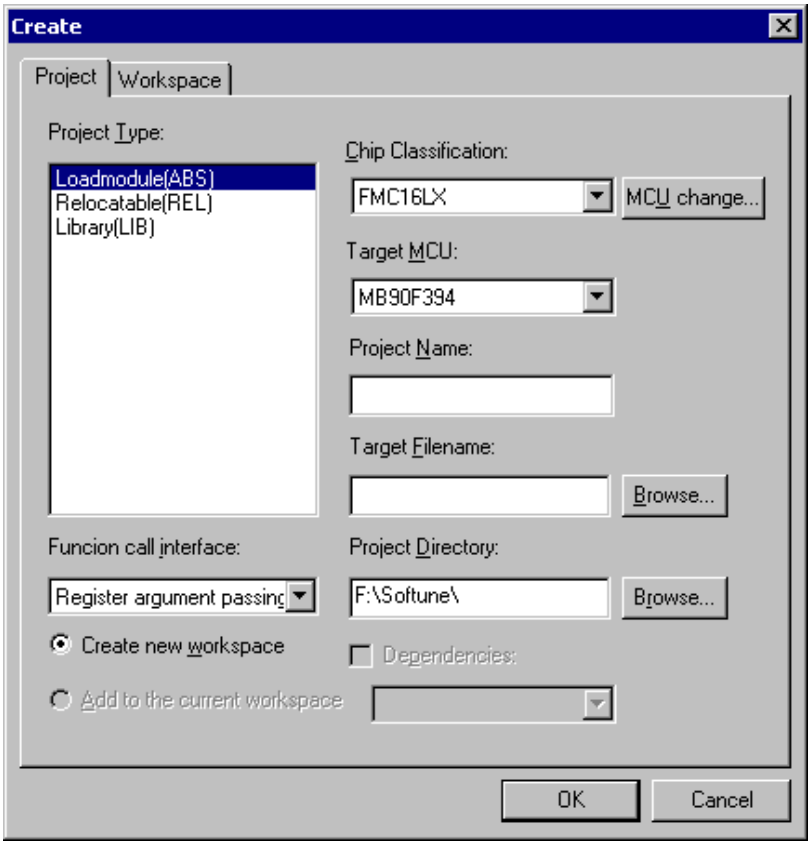


2. When changing the setting of a new project

   In the [Create] dialog of the project, "stack argument passing" is selected by default.

   Change the function-call interface to "register argument passing."

Figure H-3 Create Dialog of Project



● Changing without using Workbench

1. Changing without using Workbench

   When not using Workbench, specify the -rp option for Compiler, Assembler, Linker, and Librarian.

2. Changing during compiling and assembling

   Declare the prototypes of all functions. If a function whose prototype is not declared is called, the program is not guaranteed to operate.

   The area where argument registers are saved is not concatenated with the area where parameters area passed by the stack. A program designed to expect such concatenation does not operate properly. If a program of this type exits, it must be corrected.

   Similar to the conventional function-call interface, all arguments of the variable arguments are passed by the stack and need not to be corrected.

3. Changing during linking and making library

   The following C library is added for "register argument passing."

   When using "register argument passing," link the following library with "_rp."

   - lib/907/lib9*_rp.lib : 48 files

■ Warning and Error Messages

When a contradiction concerning the function-call interface is detected, Workbench, Assembler, Linker and Librarian output the following warning or error messages

● Workbench warning messages

- When adding of project

  • I0227W

  Projects with different function-call interfaces exist in the workspace.

  Objects created by the project with different function-call interfaces cannot be used together. Continue processing?

  YES: Add a project.

  NO: Do not add a project.

- When setting dependencies between projects (sub projects)

  •I0228W

  Function-call interfaces between projects are different.

  Objects created by the project with different function-call interfaces cannot be used together. Continue processing?

  YES: Set the dependencies between projects.

  NO: Do not set the dependencies between projects.

- When changing function-call interface

  •I0229W

  When changing the function-call interface, the function-call interfaces of the assembler source, object, and library to be used must be standardized.

  If the function-call interfaces are not standardized, the program is not guaranteed to operate.

  This change is applied to all project configurations in the project.

  Change?

  YES: Change the function-call interface.

  NO: Do not change the function-call interface.

● Assembler error message

•E4718A

Different calling interface of a function.

This error message is output under the conditions shown in Appendix Table H-1 .

Table H-1 Relationship between Description of .REG_PASS Pseudo-instructions and -rp Option

		Description of .REG_PASS pseudo-instructions	
		Provided	Unprovided
Specifying of - rp option	Provided	Normal combination for "register argument passing" No error is output.	The error (E4718A) is output.
	Unprovided	The error (E4718A) is output.	Normal combination for "stack argument passing" No error is output.

If these errors occur, correct the specifying of the -rp option or the description of the .RE_PASS pseudo-instructions for the assembler source, referring to Appendix Table H-1 .

● Linker error message

•E4313L

The module is different calling interface of a function (file name).
This error message is output under the conditions shown in Table H-2 .

Table H-2 Relationship between Input Object and -rp Option

		Function-call interface for input object	
		"register argument passing"	"stack argument passing"
Specifying of - rp option	Provided	Normal combination for "register argument passing" No error is output.	The error (E4313L) is output.
	Unprovided	The error (E4313L) is output.	Normal combination for "stack argument passing" No error is output.

If these errors occur, correct the specifying of the -rp option or the function-call interface for the input object, referring to Appendix Table H-2 .

● Librarian error message

•E4410U

The module is different calling interface of a function (file name).
This error message is output under the conditions shown in Table H-3 .

Table H-3 Relationship between Input Object and -rp Option

		Function-call interface for input object and library to be edited	
		"register argument passing"	"stack argument passing"
Specifying of - rp option	Provided	Normal combination for "register argument passing" No error is output.	The error (E4410U) is output.
	Unprovided	The error (E4410U) is output.	Normal combination for "stack argument passing" No error is output.

If these errors occur, correct the specifying of the -rp option or the function-call interface for the input object and library to be edited, referring to Table H-3 .

Note:

Objects with different function-call interfaces cannot be linked.

Fujitsu provides only the following object function-call interfaces for "stack argument passing."

- Monitor Debugger

APPENDIX I External I/F DLL for Simulator

Fast version of simulator debugger supports the external I/F to create peripheral simulation modules.
This section describes external I/F function.

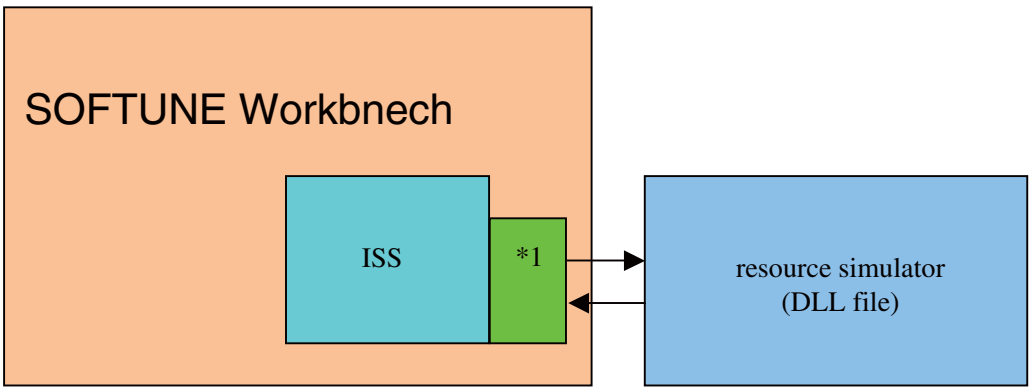
■ Outline of External I/F DLL

The Simulator Debugger for SOFTUNE Workbench supports the I/O simulation function (ports/interrupts) for aiding in debugging applications using microcontroller resources.

However, this function is simple so that it cannot cover the simulation of complicated microcontroller resources; therefore, providing interface between the instruction set simulator (ISS) and a microcontroller resource simulator that the user describes in the C language.

This microcontroller resource simulator is created by the dynamic link library (DLL) file for Windows.

■ Configuration



*1: External I/F module

The microcontroller resource simulator is read only when the Simulator Debugger for SOFTUNE Workbench is started.

The interface between the ISS and the microcontroller resource simulator is called in the following timing:

- When the Debugger is started
- When the Debugger is terminated
- When the target is reset
- Immediately before data is read from memory (I/O)
- Immediately after data is written to memory (I/O)
- Immediately before fetching is performed
- When an interrupt occurs
- When a timer event occurs

There are the following functions for operating the ISS from the microcontroller resource simulator:

- Reads/writes data from/to memory
- Reads/writes data from/to registers

- Sets interrupt sources
- Requests abort of instruction execution

■ Simulator external I/F specification

[Function List]

ISS → DLL

1. SSDI_Entry (Startup)
2. SSDI_Init (Initialize)
3. SSDI_Ready (Ready)
4. SSDI_End (End)
5. SSDI_Reset_Event (Post Reset)
6. SSDI_Read_Event (Post Read)
7. SSDI_Write_Event (Post Write)
8. SSDI_Execute_Event (Post Execute Instruction)
9. SSDI_Interrupt_Event (Post Interrupt)
10. SSDI_Timer_Event (Post Timer)

DLL → ISS

11. SSDI_Read_Memory (Read from Memory)
12. SSDI_Write_Memory (Write to Memory)
13. SSDI_Read_Register (Read from Register)
14. SSDI_Write_Register (Write to Register)
15. SSDI_Set_Interrupt (Set Interrupt Source)
16. SSDI_Set_Timer (Set Timer)
17. SSDI_Request_Abort (Request Abort)
18. SSDI_Set_Area (Set Area)
19. SSDI_Clear_Cycle (Clear cycle count)
20. SSDI_Execute_MCU (Execute target program)
21. SSDI_GetCurrentExecuteAddr (Get the address in execution)

1 SSDI_Entry (Start)

[Format]

int SSDI_Entry (int id, HINSTANCE hInstance)

[Argument]

int	id;	// DLL ID
HINSTANCE	hInstance	// External I/F module handle

[Return value]

Return NORMAL(0).

Returning any value other than NORMAL(0) causes a failure to start debugger.

[Explanation]

A call is made when the debugger is started.

Perform necessary initialization.

[Remarks]

DLL ID (id) always indicates 0.

The external I/F module handle (hInstance) is used to obtain an ISS entry.

At this point, no external I/F module can be called.

(Example)

```
typedef int (WINAPI *LPFNREADMEMORY)();
LPFNREADMEMORY Read_Memory;
extern "C" int WINAPI SSDI_Entry(int id, HINSTANCE if)
{
   int my_dll_id = id;
   FARPROC pF_Read_Memory = ::GetProcAddress(if, "SSDI_Read_Memory");
   Read_Memory = (LPFNREADMEMORY)pF_Read_Memory;

}

int func()
{
   if (ReadMemory(addr, size, len, data) != len)
                        error();

}
```

2 SSDI_Init (Initialize)

[Format]

int SSDI_Init (void)

[Argument]

None

[Return value]

Return NORMAL(0).

Returning any value other than NORMAL(0) causes a failure to start debugger starting to fail.

[Explanation]

A call is made when the debugger is started.

Perform necessary initialization.

[Remarks]

Now an external I/F module can be called.

3 SSDI_Ready (Ready)

[Format]

void SSDI_Ready (void)

[Argument]

None

[Return value]

Return NORMAL(0).

Returning any value other than NORMAL(0) causes a failure to start debugger starting to fail.

[Explanation]

When debugger activation was completed, it is called.

[Remarks]

It is called after having done batch file execution and an automatic load in debugger start.

4 SSDI_End (End)

[Format]

void SSDI_End (void)

[Argument]

None

[Return value]

None

[Explanation]

A call is made when the debugger is ended.

Perform necessary end processing.

[Remarks]

5 SSDI_Reset_Event (Post Reset)

[Format]

void SSDI_Reset_Event (void)

[Argument]

None

[Return value]

None

[Explanation]

Posts a reset of the debugger by a command or the issue of a reset.

Initializes resources.

[Remarks]

Sets all the interrupt states OFF.

6 SSDI_Read_Event (Post Read)

[Format]

int SSDI_Read_Event (READ_EVENT *info)

[Argument]

```
typedef struct {
  unsigned long     addr;          // Access address
  int               size;          // Access size (1:Byte / 2:Word / 4:LONG)
  unsigned long     total_cycle;   // Total cycle count
  unsigned long     inst_cycle;    // Count of cycles from beginning of instruction
  unsigned long     *data;         // Read data
  unsigned long     *cycle;        // Count of cycles taken for access
} READ_EVENT;
```

[Return value]

=0	Read data enabled
!=0	Read data disabled

[Explanation]

Posts the occurrence of a read access event by instruction execution.

When read data is enabled, the ISS operates assuming read data (info.data) as read data. When read data reflected in ISS memory, data must be written by using SSDI_Write_Memory().

When read data is disabled, data is read from ISS memory.

Set the count of cycles taken for read access as the count of cycles taken for access (info.cycle).

If this access cause an error, use SSDI_Request_Abort() to stop instruction execution.

[Remarks]

The count of cycles from the beginning of the instruction (inst_cycle) is always posted as 0.

7 SSDI_Write_Event (Post Write)

[Format]

int SSDI_Write_Event (WRITE_EVENT *info)

[Argument]

```
typedef struct {
  unsigned long     addr;          // Access address
  int               size;          // Access size (1:Byte / 2:Word / 4:LONG)
  unsigned long     data;          // Write data
  unsigned long     total_cycle;   // Total cycle count
  unsigned long     inst_cycle;    // Count of cycles from beginning of instruction
  unsigned long     *cycle;        // Count of cycles taken for access
} WRITE_EVENT;
```

[Return value]

=0	Write data enables
!=0	Write data disabled

[Explanation]

Posts the occurrence of a write access event by instruction execution.

When write data is enabled, the ISS does not write data to memory. Therefore, when write data is reflected in ISS memory, data must be written by using SSDI_Write_Memory().

When write data is disabled, data is written to ISS memory.

Set the count of cycles taken for write access as the count of cycles taken for access (info.cycle).

If this access cause an error, use SSDI_Request_Abort() to stop instruction execution.

[Remarks]

The count of cycles from the beginning of the instruction (inst_cycle) is always posted as 0.

8 SSDI_Execute_Event (Post Execute Instruction)

[Format]

void SSDI_Execute_Event (EXECUTE_EVENT *info)

[Argument]

```
typedef struct {
  unsigned long     addr;          // Access address
  unsigned long     total_cycle;   // Total cycle count
} EXECUTE_EVENT;
```

[Return value]

None

[Explanation]

Posts the occurrence of a fetch access event by instruction execution. This event posts only the starting address of the instruction immediately before instruction execution.

[Remarks]

9 SSDI_Interrupt_Event (Post Interrupt)

[Format]

void SSDI_Interrupt_Event (INTERRUPT_EVENT *info)

[Argument]

```
typedef struct {
  int             int_number;   // Interrupt number
  unsigned long   total_cycle;  // Total cycle count
  unsigned long   int_cycle;    // Cycle count of interrupt processing
  unsigned long   *cycle;       // Count of cycles delayed
} INTERRUPT_EVENT;
```

[Return value]

None

[Explanation]

Posts the occurrence of an interrupt event by instruction execution.

Set the count of cycles generated during processing as the count of cycles delayed (info.cycle). In the ISS, this value is added to the cycle count.

If this event causes an error, stop instruction execution by using SSDI_Request_Abort().

[Remarks]

The cycle count of interrupt processing (int_cycle) is always posted as 0.

10 SSDI_Timer_Event (Post Timer)

[Format]

void SSDI_Timer_Event (TIMER_EVENT *info)

[Argument]

```
typedef struct {
  unsigned long     total_cycle;    // Total cycle count
  unsigned long     inst_cycle;     // Differential cycle count from previous event
  int               time_id;        // Set timer ID
  unsigned long     *cycle;         // Count of cycles delayed
} TIMER_EVENT;
```

[Return value]

None

[Explanation]

Posts the occurrence of a timer event set in the ISS.

Set the count of cycles generated during processing as the count of cycles delayed (info.cycle). In the ISS, this value is added to the cycle count.

If this event causes an error, stop instruction execution by using SSDI_Request_Abort().

[Remarks]

Set the timer event conditions in SSDI_Set_Timer().

11 SSDI_Read_Memroy (Read from Memory)

[Format]

int SSDI_Read_Memory (unsigned long addr, int size, int length, void *data)

[Argument]

unsigned long	addr;	// Access address
int	size;	// Access size (1:Byte / 2:Word / 4:LONG)
int	length;	// Read data count
void	*data;	// Data storage area

[Return value]

Count of data completely read.

If the return value is different from the read data count (length), an error occurs.

[Explanation]

Reads data from memory.

The type of the data storage area (data) differs depending on the access size.

Byte (1 byte)	unsigned char []
Word (2 bytes)	unsigned short []
Long (4 bytes)	unsigned long []

[Remarks]

In this access, the reading of data is not posted.

12 SSDI_Write_Memroy (Write to Memory)

[Format]

int SSDI_Write_Memory (unsigned long addr, int size, int length, void *data)

[Argument]

unsigned long	addr;	// Access address
int	size;	// Access size (1:Byte / 2:Word / 4:LONG)
int	length;	// Write data count
void	*data;	// Data storage area

[Return value]

Count of data completely written

If the return value is different from the write data count (length), an error occurs.

[Explanation]

Writes data to memory

The type of the data storage area (data) differs depending on the access size.

Byte (1 byte)	unsigned char []
Word (2 bytes)	unsigned short []
Long (4 bytes)	unsigned long []

[Remarks]

In this access, the writing of data is not posted.

13 SSDI_Read_Register (Read from Register)

[Format]

int SSDI_Read_Register (int reg_no, unsigned long *data)

[Argument]

int	reg_no;	// Register number
unsigned long	*data;	// Data storage area

[Return value]

=0	Normal end
!=0	Error

[Explanation]

Reads data from registers

[Remarks]

The register number is defined in the include file (SSDI_REGISTER.H).

"SSDI_REGISTER.H" is installed "X:YYY\LIB\ZZZ\SSDI" folder.

X:YYY	It is the drive & folder which it installed SOFTUNE in
ZZZ	CPU family name is used there. (F^2MC-16 is 907)

14 SSDI_Write_Register (Write to Register)

[Format]

int SSDI_Write_Register (int reg_no, unsigned long data)

[Argument]

int	reg_no;	// Register number
unsigned long	data;	// Data storage area

[Return value]

=0	Normal end
!=0	Error

[Explanation]

Writes data to registers

[Remarks]

The register number is defined in the include file (SSDI_REGISTER.H).

"SSDI_REGISTER.H" is installed "X:YYY\LIB\ZZZ\SSDI" folder.

X:YYY	It is the drive & folder which it installed SOFTUNE in
ZZZ	CPU family name is used there. (F^2MC-16 is 907)

15 SSDI_Set_Interrupt (Set Interrupt Source)

[Format]

int SSDI_Set_Interrupt (int int_no, int sw)

[Argument]

int	int_no;	// Interrupt number
int	sw;	// Interrupt state (=0:OFF / =1:ON)

[Return value]

=0	Normal end
!=0	Error

[Explanation]

Sets the interrupt state.

If an interrupt is accepted, the call back (SSDI_Interrupt_Event) is called.

[Remarks]

If the interrupt state is set ON, an interrupt request always occurs. Set the interrupt state OFF in the timing in which the interrupt source flag is cleared.

16 SSDI_Set_Timer (Set Timer)

[Format]

int SSDI_Set_Timer (int no, unsigned long cycle, int sw, int id)

[Argument]

int	no;	// Timer setting number
unsigned long	cycle;	// Cycle count
int	sw;	// Condition (0: Repeat/1: Only once)
int	id;	// Always set 0

[Return value]

Timer setting numbers (0 to 31)

=-1 Error

[Explanation]

Generate a timer event after an elapse of the set cycle count after the timer is set.

There are two timer setting conditions, repeat and only once.

To clear the timer setting, set the cycle count (cycle) to - 1.

If the timer setting number is set to -1, it is set to an unassigned number.

[Remarks]

17 SSDI_Request_Abort (Request Abort)

[Format]

void SSDI_Request_Abort (char *message)

[Argument]

char	*message;	// Abort message

[Return value]

None

[Explanation]

Requests the ISS to abort.

When an abort is requested, the ISS aborts processing when the current instruction execution is terminated.

The abort message (message) is displayed as the abort message for the debugger.

[Remarks]

18 SSDI_Set_Area (Set Area)

[Format]

int SSDI_Set_Area (int no, unsigned long start, unsigned long size, int attribute, int id)

[Argument]

int	no;	// Area setting number
unsigned long	start;	// Starting address of area
unsigned long	size;	// Area size
unsigned long	attribute;	// Attribute
int	id;	// Always set 0

[Return value]

Area setting numbers (0 to 31)

=-1 Error

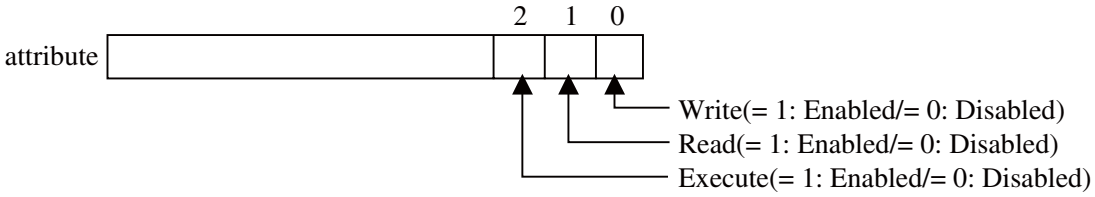
[Explanation]

Sets the area where a read event, write event or execute instruction event occurs.

An event occurs only upon access to the area in which an area is set. Set an area by using SSDI_Init (Initialize).

If the area setting number is set to -1, the area setting address/area size is valid. In other cases, set the attributes for the settings in the area setting numbers.

Set the attributes as follows:



To delete the area settings, set the number of the area to be deleted in the area setting number and the attribute to 0.

Up to 32 areas can be set. If the maximum value is exceeded, an error occurs.

[Remarks]

19 SSDI_Clear_Cycle (Clear cycle count)

[Format]

int SSDI_Clear_Cycle (void)

[Argument]

None

[Return value]

None

[Explanation]

Initialize total cycle number managing with ISS(Instruction Set Simulator).

[Remarks]

20 SSDI_Execute_MCU (Execute target program)

[Format]

void SSDI_Execute_MCU (void)

[Argument]

None

[Return value]

None

[Explanation]

Starts execution of program.

When execution of program starts it already, it is ignored.

[Remarks]

21 SSDI_GetCurrentExecuteAddr (Get the Address in Execution)

[Format]

unsigned long SSDI_GetCurrentExecuteAddr (void)

[Argument]

None

[Return value]

The address of currently executed instruction

[Explanation]

Get the address of the instruction currently in execution.

[Remarks]

CM41-00312-3E

FUJITSU SEMICONDUCTOR • CONTROLLER MANUAL

F^2MC-16 FAMILY
SOFTUNE™ WORKBENCH
OPERATION MANUAL

January 2007 the third edition

Published **FUJITSU LIMITED** Electronic Devices

Edited Business Promotion Dept.